

Miami

COLLABORATORS

	<i>TITLE :</i> Miami		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 10, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Miami	1
1.1	Miami.guide	1
1.2	Miami.guide/NODE_DISCLAIMER	2
1.3	Miami.guide/NODE_CONDITIONS	4
1.4	Miami.guide/NODE_REGISTRATION	6
1.5	Miami.guide/NODE_INTRODUCTION	6
1.6	Miami.guide/NODE_REQUIREMENTS	8
1.7	Miami.guide/NODE_INSTALLATION	9
1.8	Miami.guide/NODE_MIAMIINIT	10
1.9	Miami.guide/NODE_TOOLTYPES	10
1.10	Miami.guide/NODE_MENU	12
1.11	Miami.guide/NODE_CONFIGURATION	13
1.12	Miami.guide/NODE_GUI_GENERAL	14
1.13	Miami.guide/NODE_GUI_GENERAL_REGISTER	14
1.14	Miami.guide/NODE_GUI_INTERFACE	14
1.15	Miami.guide/NODE_GUI_INTERFACE_TYPE	16
1.16	Miami.guide/NODE_GUI_INTERFACE_DEVICE	17
1.17	Miami.guide/NODE_GUI_INTERFACE_SPEED	17
1.18	Miami.guide/NODE_GUI_INTERFACE_CD	18
1.19	Miami.guide/NODE_GUI_INTERFACE_PROTOCOL	18
1.20	Miami.guide/NODE_GUI_INTERFACE_FLOW	19
1.21	Miami.guide/NODE_GUI_INTERFACE_EOF	19
1.22	Miami.guide/NODE_GUI_INTERFACE_SERIAL	19
1.23	Miami.guide/NODE_GUI_INTERFACE_MNIOPT	20
1.24	Miami.guide/NODE_GUI_INTERFACE_IP	20
1.25	Miami.guide/NODE_GUI_INTERFACE_MASK	21
1.26	Miami.guide/NODE_GUI_INTERFACE_GWAY	21
1.27	Miami.guide/NODE_GUI_INTERFACE_MULTICASTS	22
1.28	Miami.guide/NODE_GUI_INTERFACE_MAPPING	22
1.29	Miami.guide/NODE_GUI_INTERFACE_MTU	23

1.30	Miami.guide/NODE_GUI_INTERFACE_STP	24
1.31	Miami.guide/NODE_GUI_INTERFACE_MNIP	24
1.32	Miami.guide/NODE_GUI_INTERFACE_MNIINFO	25
1.33	Miami.guide/NODE_GUI_INTERFACE_FINDB	25
1.34	Miami.guide/NODE_GUI_INTERFACE_INACTIVITY	25
1.35	Miami.guide/NODE_GUI_PPP	26
1.36	Miami.guide/NODE_GUI_PPP_CHAP	27
1.37	Miami.guide/NODE_GUI_PPP_CALLBACK	28
1.38	Miami.guide/NODE_GUI_PPP_VJC	28
1.39	Miami.guide/NODE_GUI_PPP_ACCM	28
1.40	Miami.guide/NODE_GUI_PPP_QUICK	29
1.41	Miami.guide/NODE_GUI_PPP_ESCAPE	30
1.42	Miami.guide/NODE_GUI_PPP_DNSIPCP	30
1.43	Miami.guide/NODE_GUI_PPP_TERMREQ	30
1.44	Miami.guide/NODE_GUI_DIALER	31
1.45	Miami.guide/NODE_GUI_DIALER_SCRIPT	31
1.46	Miami.guide/NODE_GUI_DIALER_PHONE	32
1.47	Miami.guide/NODE_GUI_DIALER_MAX	32
1.48	Miami.guide/NODE_GUI_DIALER_DELAY	32
1.49	Miami.guide/NODE_GUI_DIALER_RDELAY	33
1.50	Miami.guide/NODE_GUI_DIALER_TEACH	33
1.51	Miami.guide/NODE_GUI_DIALER_NAME	33
1.52	Miami.guide/NODE_GUI_DIALER_CAPTURE	34
1.53	Miami.guide/NODE_GUI_DATABASE	34
1.54	Miami.guide/NODE_GUI_DATABASE_PROTOCOLS	35
1.55	Miami.guide/NODE_GUI_DATABASE_SERVICES	36
1.56	Miami.guide/NODE_GUI_DATABASE_HOSTS	36
1.57	Miami.guide/NODE_GUI_DATABASE_NETWORKS	36
1.58	Miami.guide/NODE_GUI_DATABASE_DOMAINS	37
1.59	Miami.guide/NODE_GUI_DATABASE_DNSSERVERS	37
1.60	Miami.guide/NODE_GUI_DATABASE_INETD	38
1.61	Miami.guide/NODE_GUI_DATABASE_USERS	38
1.62	Miami.guide/NODE_GUI_DATABASE_GROUPS	39
1.63	Miami.guide/NODE_GUI_DATABASE_ARP	40
1.64	Miami.guide/NODE_GUI_DATABASE SOCKS	40
1.65	Miami.guide/NODE_GUI_DATABASE_IPFILTER	41
1.66	Miami.guide/NODE_GUI_TCPIP	43
1.67	Miami.guide/NODE_GUI_TCPIP_HOSTNAME	44
1.68	Miami.guide/NODE_GUI_TCPIP_NAME	44

1.69	Miami.guide/NODE_GUI_TCPIP_ICMP	44
1.70	Miami.guide/NODE_GUI_TCPIP_BOOTP	45
1.71	Miami.guide/NODE_GUI_TCPIP_VERIFYDNS	45
1.72	Miami.guide/NODE_GUI_TCPIP_FAKEIP	45
1.73	Miami.guide/NODE_GUI_TCPIP_TTCP	46
1.74	Miami.guide/NODE_GUI_TCPIP_ADDDOMAIN	46
1.75	Miami.guide/NODE_GUI_TCPIP_DOWN	46
1.76	Miami.guide/NODE_GUI_TCPIP_PING	47
1.77	Miami.guide/NODE_GUI_TCPIP_GETTIME	47
1.78	Miami.guide/NODE_GUI_EVENTS	48
1.79	Miami.guide/NODE_GUI_MODEM	49
1.80	Miami.guide/NODE_GUI_MODEM_INIT	50
1.81	Miami.guide/NODE_GUI_MODEM_EXIT	50
1.82	Miami.guide/NODE_GUI_MODEM_PREFIX	50
1.83	Miami.guide/NODE_GUI_MODEM_SUFFIX	50
1.84	Miami.guide/NODE_GUI_MODEM_NULLMODEM	50
1.85	Miami.guide/NODE_GUI_LOGGING	51
1.86	Miami.guide/NODE_GUI_LOGGING_CONSOLE	51
1.87	Miami.guide/NODE_GUI_LOGGING_FILE	52
1.88	Miami.guide/NODE_GUI_LOGGING_SYSLOG	52
1.89	Miami.guide/NODE_GUI_LOGGING_PHONE	52
1.90	Miami.guide/NODE_GUI_LOGGING_PPP	53
1.91	Miami.guide/NODE_GUI_WINDOWS	53
1.92	Miami.guide/NODE_GUI_WINDOWS_REQQUIT	53
1.93	Miami.guide/NODE_GUI_WINDOWS_REQOFFLINE	54
1.94	Miami.guide/NODE_GUI_WINDOWS_REQERRORS	54
1.95	Miami.guide/NODE_GUI_WINDOWS_DIALER	54
1.96	Miami.guide/NODE_GUI_GUI	55
1.97	Miami.guide/NODE_GUI_GUI_HOTKEY	55
1.98	Miami.guide/NODE_GUI_GUI_SHOWICON	56
1.99	Miami.guide/NODE_GUI_GUI_SHOWMENU	56
1.100	Miami.guide/NODE_GUI_GUI_ONSTARTUP	56
1.101	Miami.guide/NODE_GUI_GUI_ONLINEICON	56
1.102	Miami.guide/NODE_GUI_GUI_OFFLINEICON	57
1.103	Miami.guide/NODE_GUI_GUI_GUI	57
1.104	Miami.guide/NODE_GUI_GUI_SWITCH	57
1.105	Miami.guide/NODE_GUI SOCKS	57
1.106	Miami.guide/NODE_GUI SOCKS_ENABLE	58
1.107	Miami.guide/NODE_GUI SOCKS_SERVER	58

1.108Miami.guide/NODE_GUI SOCKS_MAXLOG	59
1.109Miami.guide/NODE_GUI SOCKS_AUTH	59
1.110Miami.guide/NODE_GUI_MISC	59
1.111Miami.guide/NODE_MNI	60
1.112Miami.guide/NODE_MNI_GGTDPETNZ	61
1.113Miami.guide/NODE_MNI_ZTAMSNNZ	62
1.114Miami.guide/NODE_MNI_ZTDPETNZ	62
1.115Miami.guide/NODE_MNI_ZTMBESNFZ	63
1.116Miami.guide/NODE_MNI_ZTSMCNOCNZ	64
1.117Miami.guide/NODE_DIALERLANG	64
1.118Miami.guide/NODE_AREXX	66
1.119Miami.guide/NODE_ENVVARS	67
1.120Miami.guide/NODE_ENVVARS_TZ	69
1.121Miami.guide/NODE_EXCONFIG	69
1.122Miami.guide/NODE_EXCONFIG_DIST	70
1.123Miami.guide/NODE_EXCONFIG_PASSWORDS	73
1.124Miami.guide/NODE_EXCONFIG_CLIENTS	74
1.125Miami.guide/NODE_UTILITY	75
1.126Miami.guide/NODE_UTILITY_ARP	76
1.127Miami.guide/NODE_UTILITY_FINGER	77
1.128Miami.guide/NODE_UTILITY_IFCONFIG	77
1.129Miami.guide/NODE_UTILITY_MAPMBONE	79
1.130Miami.guide/NODE_UTILITY_MRINFO	79
1.131Miami.guide/NODE_UTILITY_MROUTED	80
1.132Miami.guide/NODE_UTILITY_MTRACE	81
1.133Miami.guide/NODE_UTILITY_NETSTAT	83
1.134Miami.guide/NODE_UTILITY_PING	86
1.135Miami.guide/NODE_UTILITY_REMIND	88
1.136Miami.guide/NODE_UTILITY_RESOLVE	90
1.137Miami.guide/NODE_UTILITY_ROUTE	90
1.138Miami.guide/NODE_UTILITY_SYSCTL	92
1.139Miami.guide/NODE_UTILITY_TCPDUMP	95
1.140Miami.guide/NODE_UTILITY_TRACEROUTE	97
1.141Miami.guide/NODE_COMPATIBILITY	99
1.142Miami.guide/NODE_RESTRICTIONS	99
1.143Miami.guide/NODE_HISTORY	100
1.144Miami.guide/NODE_FUTURE	101
1.145Miami.guide/NODE_SUPPORT	101
1.146Miami.guide/NODE_ACKNOWLEDGEMENTS	102

Chapter 1

Miami

1.1 Miami.guide

Miami

Det här är dokumentationen för Miami v3.2, ett integrerat TCP/IP-system för AmigaOS. Copyright (C) 1996-1998 Nordic Global Inc. Alla rättigheter reserverade. Program och dokumentation av Holger Kruse.

Frånsägande av ansvaret

Juridisk information

Användning/kopiering

Krav för användning/kopiering

Registrering

Sharewareregistrering

Introduktion

Introduktion till Miami

Systemkrav

Hård- och mjukvarukrav

Installation

Hur du installerar Miami

MiamiInit

Snabbstart med MiamiInit

Verktygstyper

Verktygstyper för Miami

Menyer

Miamis menyer

Inställningar

	Hur du konfigurerar Miami
MNI-drivrutiner	MNI-drivrutiner för Ethernet
Språk för uppringaren	Beskrivning av uppringaren
ARexxgränssnitt	ARexx-kommandon som stöds
Miljövariabler	Miljövariabler
Byta inställningar	Import/export av inställningar
Nyttoprogram	Program som följer med Miami
Kompatibilitet	Kompatibilitetsfrågor
Begränsningar	Programmets begränsningar
Historia	Miamis historia
I framtiden	Miamis framtid
Support	Hur du får hjälp
Tack	Tack till olika personer

1.2 Miami.guide/NODE_DISCLAIMER

Frånsägande av ansvaret

Miami IS SUPPOSED TO BE A TCP/IP PACKAGE FOR AmigaOS THAT CAN BE USED TO CONNECT YOUR AMIGA TO THE INTERNET BY MODEM OR THROUGH A NETWORK DEVICE. EVEN THOUGH EVERY EFFORT HAS BEEN MADE TO MAKE Miami AS COMPATIBLE TO THE TCP/IP STANDARD AS POSSIBLE, I CANNOT RULE OUT THE POSSIBILITY THAT Miami HAS BUGS THAT HAVE HARMFUL SIDE EFFECTS ON YOUR SYSTEM OR ON OTHER MACHINES CONNECTED TO YOUR AMIGA.

I HEREBY REJECT ANY LIABILITY OR RESPONSIBILITY FOR THESE OR ANY OTHER CONSEQUENCES FROM THE USE OF Miami WHATSOEVER. THIS INCLUDES, BUT IS NOT LIMITED TO, DAMAGE TO YOUR EQUIPMENT, TO YOUR DATA, TO OTHER

MACHINES YOUR AMIGA IS CONNECTED TO, ANY EQUIPMENT CONNECTED TO THOSE HOSTS, PERSONAL INJURIES, FINANCIAL LOSS OR ANY OTHER KINDS OF SIDE EFFECTS.

Miami IS PROVIDED AS-IS. THIS MEANS I DO NOT GUARANTEE THAT Miami IS FIT FOR ANY SPECIFIC PURPOSE AND I DO NOT GUARANTEE ANY BUG FIXES, UPDATES OR HELP DURING ERROR RECOVERY.

Miami bygger på nätverkskällkoden till 4.4BSD V.2 TCP/IP, den version som distribueras av Walnut Creek på cd-rom.

Hela den ursprungliga 4.4BSD-koden är fritt tillgänglig och har gjorts av flera personer. För detaljerad information om individuella regler om upphovsrätt och fränsägande av ansvaret, se källkoden som är tillgänglig från olika källor, till exempel från 4.4BSD Lite som finns på cd-rom hos Walnut Creek.

Följande upphovsrättsmeddelande gäller hela mjukvarupaketet 4.4BSD:

Citat

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.

Slut citat

Observera att ovanstående upphovsrättsmeddelande inte gäller för Miami. Miami är inte fritt tillgängligt, om det inte står så. Se

Användning/kopiering
för mer information.

Några av Miamis GUI-motorer kräver Magic User Interface (MUI). MUI är Copyright Stefan Stuntz.

Några av Miamis GUI-motorer kräver MUI-klassen "Busy.mcc" av Klaus Melchior. Här är det tillhörande upphovsrättsmeddelandet:

Citat

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

Slut citat

Några av Miamis GUI-motorer kräver gtlayout.library av Olaf Barthel. Här är det tillhörande upphovsrättsmeddelandet:

Citat

Copyright © 1993-1996 by Olaf 'Olsen' Barthel Freely distributable.

Slut citat

Några av Miamis GUI-motorer använder bilder baserade på MagicWB som har kopierats med tillåtelse av skaparen. Här är det tillhörande upphovsrättsmeddelandet. Mer information finns i filen MagicWB.readme i Miami-distributionen.

Citat

Copyright © 1992-97, Martin Huttenloher

Slut citat

1.3 Miami.guide/NODE_CONDITIONS

Användning/kopiering

Miami är shareware. I det här fallet betyder det att en personlig nyckelfil krävs för att du ska kunna använda alla funktioner i Miami.

Användarna kommer att få sina personliga nyckelfiler från mig efter att de har registrerat sig. Nyckelfilen får inte göras tillgänglig för andra användare! Att ge nyckelfilen till någon annan eller att för personligt bruk använda nyckelfiler som du inte fått direkt från mig betraktas som piratkopiering!

Nyckelfiler kan ej överlåtas och får ej säljas till eller bytas

med en annan person eller organisation. De är avsedda att endast användas av personen som registrerat sig.

Binärfilen till Miami eller binärfilerna till något av de andra programmen får inte ändras eller "patchas" på något sätt (inte ens för personligt bruk) förutom i situationer som godkänns av mig vid uppdateringar. Att använda "patchade" eller ändrade binärfiler betraktas som piratkopiering.

Binärfiler till Miami får endast användas för det de är ämnade till, det vill säga, att köras på Amigadatorer under AmigaOS. Att bakåttassemblera, utföra "reverse-engineering" eller översätta binärfiler är uttryckligen förbjudet.

Dokumentationen och programtexterna till Miami lyder under samma upphovsrättsregler som programmet. Det betyder att varken dokumentationen eller programtexterna får ändras eller översättas på något sätt.

För att undvika missförstånd: Du får inte översätta och distribuera Miamis programtexter och dokumentation, om inte jag officiellt utser dig till översättare av Miami. Otillåtna översättningar av programtexterna eller dokumentationen är olagliga, strider mot mina upphovsrättsregler, och kommer att tas bort från allmänna mjukvaruplatser.

Vill du distribuera Miamiarkivet måste följande krav gälla:

- * Försäljningspriset får inte överstiga kostnaden för de tomma disketter som krävs för Miami, en obetydlig kopieringsavgift och fraktkostnader. Det totala priset får ej överstiga 10 US-dollar eller 15 DM eller motsvarigheten i någon annan valuta.
 - * Om Miamiarkivet kommer att distribueras som en del av en cd-romsamling med public domain- eller sharewareprogram får inte försäljningspriset för cd:n överstiga 20 US-dollar eller 30 DM eller motsvarigheten i någon annan valuta.
 - * Alla delar av programmet och dokumentationen måste vara fullständiga. Distribution av bara vissa delar eller ofullständiga delar av originaldistributionen är inte tillåtet. Distribution av nyckelfiler är inte tillåtet.
 - * Miami eller delar av det får vanligtvis inte säljas i kombination med eller som en del av ett kommersiellt program. Separata licenskrav för kommersiell återförsäljning kan fås vid förfrågan från kruse@nordicglobal.com. Anta inte att du får distribuera Miami eller delar av det i kombination med eller som del av ett kommersiellt program innan du har fått ett skriftligt godkännande av mig.
 - * Program och dokumentation får inte ändras på något sätt. Undantaget (läs: acceptabelt) är användning av packningsprogram såsom LHA så länge det är möjligt att återfå originalprogrammen/originaldatan.
-

1.4 Miami.guide/NODE_REGISTRATION

Registrering

Om du ofta använder Miami, behöver någon av de funktionerna som är avstängda i demoversionen eller vill kunna vara uppkopplad i mer än en timme i taget, rekommenderar jag att du registrerar Miami.

Kör programmet MiamiRegister för att registrera dig. Det förklarar registreringsproceduren i detalj och låter dig registrera interaktivt.

Kontakta kruse@nordicglobal.com om du av någon anledning inte kan köra registreringsprogrammet MiamiRegister.

Registreringsavgiften är 35 US-dollar för en vanlig, "full" Miami-licens. Registrerade användare av ppp.device får rabatt när de uppgraderar till Miami. Detaljerna förklaras i MiamiRegister.

Specialerbjudanden för grupplicensiering (tio användare eller mer per gång), förbetalda licenser och kommersiell försäljning av licenser förekommer. Kontakta kruse@nordicglobal.com för mer information.

Användare som redan registrerat Miami 2.x kommer att behöva nya nyckelfiler för Miami 3.x. Uppgraderingspolicyn är följande:

- * Om du fick din nyckelfil till Miami efter den 15:e juni 1997 är uppgraderingen till Miami 3.x gratis.
- * Om du fick din nyckelfil till Miami före den 15:e juni 1997 behöver du en uppgraderingskod för att kunna uppgradera dina nyckelfiler. Uppgraderingskoder finns tillgängliga från Nordic Global Inc. (för 12,00 US-dollar) och från en del registreringsställen.

Kör MiamiRegister för att uppgradera dina nyckelfiler. Programmet kommer att erbjuda uppgraderingsförslag och kommer även att berätta om du behöver betala för uppgraderingen.

Om du redan har nyckelfiler för 3.0 behöver du inte uppgradera igen för 3.2. Nyckelfiler för 3.0 fungerar för 3.2 utan någon ändring.

1.5 Miami.guide/NODE_INTRODUCTION

Introduktion

Miami är ett integrerat TCP/IP-system för AmigaOS som gör det möjligt att mycket enkelt kunna koppla upp sig mot Internet eller mot ett lokalt nätverk med modem eller med någon annan nätverksenhet, till exempel Ethernet.

Miami bygger på senaste versionen av den officiella nätverkskällkoden till BSD, 4.4BSD V2, plus några tillägg gjorda av tredjepartstillverkare, såsom "FreeBSD T/TCP" och "Path MTU discovery code". Det betyder att Miami innehåller en "riktig" och fullständig TCP/IP-stack, inte bara en emulering som endast innehåller delar av TCP/IP-standarden.

Gränssnittet för applikationsprogrammerare till Miami, API:t, är kompatibelt med det för AmiTCP 4.x (bsdsocket.library), det vill säga, de flesta program som är skrivna och kompilerade för AmiTCP 4.x kommer att fungera med Miami utan att behöva ändras eller omkompileras.

Miami har dessutom en inbyggd uppringare som både kan användas interaktivt och i olika skript, en implementering av (C)SLIP- och PPP-protokollen, ett gränssnitt för SANA-II-drivrutiner, ett grafiskt användargränssnitt för att kunna kontrollera och konfigurera programmet, en klient för SOCKS-serverar (proxyservrar) och många andra funktioner.

Miami har även en inbyggd implementering av Inetd, "Internet super-server", med många inbyggda tjänster som "fingerd" och "identd", en inbyggd implementering av TCP:, AmigaDOS strömhanterare för TCP/IP och en inbyggd implementering av usergroup.library, ett gränssnitt för att hantera användare och användargrupper.

Till skillnad från andra vanliga protokollstackar har Miami ett väl utvecklat stöd för modemuppringda förbindelser med Internet. Inställningsprocessen har gjorts så enkel som möjligt: de flesta inställningarna bestäms automatiskt av Miami. Miami stöder också fördefinierade inställningar som kan distribueras av Internetleverantörer. Miami kan också användas med andra uppkopplingar än modem, till exempel Ethernet, Arcnet eller ett kabelmodem.

Miami stöder flera olika GUI-motorer för inställningsprogrammet. För att kunna kontrollera Miami (koppla upp eller ned, eller ändra inställningarna) måste en GUI-motor laddas. Så fort som Miami är uppkopplat kan du ta bort GUI-motorn ur minnet för att spara minne. Du kan när som helst ladda in en GUI-motor i minnet igen om du vill ändra dina inställningar.

Miami stöder för närvarande följande GUI-motorer:

MUI

Den här motorn kräver MUI (Magic User Interface) 3.8 eller högre, och skapar ett användargränssnitt i typisk MUI-stil.

MUIMWB

Den här motorn är identisk med MUI-motorn, förutom att den även visar bilder på vissa ställen, inte enbart text, och är mer omsorgsfullt utformad. Den kräver MUI 3.8 eller högre och en skärm med minst 8 färger som använder MagicWB-färgpaletten.

GTLayout

Den här motorn skapar ett GadTools-baserat användargränssnitt

och kräver Olaf Barthels gtlayout.library version 40 eller högre. Den kräver inte MUI.

Den rekommenderade GUI-motorn är "MUI" eller, för färgrika skärmar, "MUIMWB". GTLayout-motorn kan även användas, men några av funktionerna i MUI-motorn kan då inte användas, till exempel sortering av databasposterna med hjälp av drag och släpp.

Innan du startar Miami bör du titta på

MiamiInit

. MiamiInit är

ett program som för de flesta användare automatiskt konfigurerar Miami för dina behov, det vill säga, uppringningsskript, autentisering, IP-adress, DNS-servrar, nätmask och alla andra inställningar.

Efter att ha kört MiamiInit ska du köra Miami, importera inställningarna från MiamiInit, spara de nya inställningarna och koppla upp dig mot din Internetleverantör.

Om du vill använda Miami med en lokal Ethernetuppkoppling vill du kanske konfigurera Miami manuellt utan hjälp av MiamiInit. Du kan använda MiamiInit men programmet stöder för närvarande inte de nya MNI-drivrutinerna. Om du använder MiamiInit kan du bara använda SANA-II-drivrutiner med ditt Ethernetkort. Det kommer att ändras i en framtida version.

1.6 Miami.guide/NODE_REQUIREMENTS

Systemkrav

För att använda Miami behöver du:

- * en Amiga som kör OS 2.04 eller högre.
- * MUI 3.8 eller högre om du vill använda en av MUI-motorerna, eller gtlayout.library v40 eller högre för GTLayout-motorn.

Du kommer också behöva hårdvara för nätverk och en dator att koppla upp dig mot. Detta kan exempelvis vara:

- * ett modem kopplat till din Amiga och till telefonlinjen. Modemet bör vara åtminstone hyfsat Hayeskompatibelt. De flesta nuvarande modem är det. Du behöver även ett SLIP- eller PPP-konto hos en Internetleverantör. Om du bara har ett "shellkonto" kan du använda Miami ändå, men då måste du installera Slirp eller TIA hos din Internetleverantör först. I sådana fall måste du fråga din Internetleverantör om du får göra det och hur och var du kan få tag på Slirp eller TIA.
 - * ett Ethernetkort, ett kabelmodem och ett SLIP/PPP-konto som beskrivs ovan.
-

* ett Ethernetkort direkt anslutet till ett lokalt nätverk.

Observera att Miami inte kräver ppp.device, appp.device, amippp.device eller (rh)(c)slip.device. PPP- och (C)SLIP-protokollen är inbyggda i Miami. De är mer effektiva och mer avancerade än de som för närvarande finns i olika SANA-II-drivrutiner.

1.7 Miami.guide/NODE_INSTALLATION

Installation

Miami kommer i följande arkiv:

Miami30main.lha

Huvudarkivet. Alla behöver detta.

Miami30-000.lha

68000/010-versionen av Miami. Du behöver det här arkivet om din Amiga har en CPU av typen 68000 eller 68010.

Miami30-020.lha

68020+-versionen av Miami. Du behöver det här arkivet om din Amiga har en CPU av typen 68020, 68030, 68040 eller 68060.

Miami30-MUI.lha

MUI-motorn för Miami. Du behöver det här arkivet om du vill använda Miami tillsammans med MUI 3.8 eller högre.

Miami30-GTL.lha

GTLayout-motorn för Miami. Du behöver det här arkivet om du vill använda Miami tillsammans med gtlayout.library v40 eller högre.

Alla måste ladda ned huvudarkivet, ett av de två CPU-arkiven (000 eller 020) och åtminstone ett av de två GUI-motorarkiven. Du kan installera mer än en GUI-motor om du vill.

Ladda ned alla arkiv du behöver, packa upp dem i samma, temporära, låda och starta installationsskriptet i den lådan för att installera Miami. Installationsskriptet kan användas för en ny installation eller för uppdateringar.

Alla filer kopieras från installationslådan till en enda låda och inga systemfiler eller -lådor kommer att röras, med ett undantag:

Installationsskriptet frågar dig om du vill skapa tilldelningen "Miami:" och lägger sedan till de nödvändiga anropen till filen User-Startup. Detta är nödvändigt. Om du hoppar över detta steg under installationen måste du efteråt skapa tilldelningen manuellt innan du startar Miami. Annars kommer Miami inte att fungera ordentligt.

1.8 Miami.guide/NODE_MIAMIINIT

MiamiInit

MiamiInit är ett program som försöker bestämma alla inställningar för Miami som behövs för en seriell uppkoppling (SLIP eller PPP) eller en SANA-II-uppkoppling (Ethernet, Arcnet, etc.) och sedan spara en inställningsfil som senare kan användas av Miami.

Det första du ska göra för att konfigurera Miami efter installationen är att köra MiamiInit och gå igenom dialogen. MiamiInit kommer under dialogens gång att koppla upp dig mot din Internetleverantör och bestämma alla nödvändiga inställningar och slutligen spara dem till disk.

MiamiInit stöder för närvarande bara de vanligaste utrustningarna. Mycket ovanliga fall, som dataformat andra än 8N1, ej Hayeskompatibla modem eller 3-trådars modemkablar stöds inte. Har du någon sådan ovanlig utrustning måste du konfigurera Miami manuellt istället för att köra MiamiInit.

Observera att beroende på konfigurationen av ditt nätverk är det inte alltid säkert att MiamiInit hittar all nödvändig information. Det har inte uppstått något fel om MiamiInit frågar dig om saker som IP-adresser eller nätmask under konfigurationen. Det betyder bara att det inte finns en server på nätverket som tillhandahåller sådan information till MiamiInitSANA2. I det fallet måste du fråga din Internetleverantör eller nätverksadministratör om informationen som fattas.

Generellt sett måste du om du sätter upp ett mycket litet lokalt nätverk som bara består av Amigor och PC:ar vanligtvis skriva in den mesta informationen för hand. Om du å andra sidan är uppkopplad mot Internet eller mot ett existerande företagsnätverk som är inställt för att konfigurera nya datorer kan MiamiInit ofta hitta det mesta eller all information på servern.

MiamiInit stöder för närvarande inte MNI-drivrutiner. Om du vill konfigurera Miami för Ethernet måste du antingen använda MiamiInit och konfigurera det för SANA-II eller konfigurera Miami manuellt för MNI. Givetvis kan du även först köra MiamiInit och konfigurera systemet för SANA-II och sedan byta till MNI efter att ha importerat inställningarna till Miami.

1.9 Miami.guide/NODE_TOOLTYPES

Verktygstyper

Miami stöder följande verktygstyper när det startas från Workbench (eller argument om det startas från Shell):

PACKETDEBUG

Initierar felsökningsläge på paketenivån. Om du anger "PACKETDEBUG=10" eller "PACKETDEBUG=20" skapar Miami en fil kallad Miami.debug innehållande en hexadecimal dumpning av alla skickade och mottagna paket. Du ska bara använda den här funktionen när du felsöker, inte under normala omständigheter, eftersom dessa loggfiler kan bli mycket stora mycket snabbt och kräver mycket CPU-tid. Om du anger värdet 10 loggar du bara paketbelastningen. Värdet 20 loggar även rå paketdata (för PPP/SLIP).

DONTCONNECT

Om du har konfigurerat Miami att automatiskt ansluta sig till din Internetleverantör varje gång du startar programmet kan du använda den här verktygstypen för att upphäva den funktionen; detta ger dig en chans att ändra inställningarna innan du kopplar upp dig.

SETTINGS

Varje projektikon måste innehålla verktygstypen "SETTINGS" så att Miami känner igen att det är en inställningsfil. Från shellfönstret kan du använda argumentet "SETTINGS=filnamn" för att välja vilken inställningsfil som ska laddas.

IMPORTMIAMIINIT

Argumentet "IMPORTMIAMIINIT=filnamn" talar om för Miami att det ska importera en inställningsfil från MiamiInit.

IMPORTASCII

Argumentet "IMPORTASCII=filnamn" talar om för Miami att det ska importera en ASCII-inställningsfil (distributionsformat).

SAVESETTINGS

Argumentet "SAVESETTINGS" talar om för Miami att det ska spara inställningarna som förvalda. Det här argumentet är mest användbart när det kombineras med "IMPORTMIAMIINIT" eller "IMPORTASCII" för att importera en främmande inställningsfil och konvertera den till en inställningsfil för Miami.

AREXX

Argumentet "AREXX=filnamn" talar om för Miami att det ska köra det specificerade AREXXskriptet när programmet startas.

PUBSCREEN

Argumentet "PUBSCREEN=namn" bestämmer vilken allmän skärm som Miami ska öppnas på. Observera att MUI-motorerna har sina egna sätt att konfigurera skärmar; detta görs genom MUI.

GUI

Argumentet "GUI=namn" talar om för Miami vilken GUI-motor som ska användas för användargränssnittet. Det här argumentet upphäver valet i inställningsfilen.

NOGUI

Argumentet "NOGUI" gör att Miami startas utan att öppna sitt grafiska gränssnitt.

Försök inte att använda odokumenterade verktygstyper! Sådana

verktygstyper gör ofta inte vad man tror att de ska göra och kan leda till reducerad kompatibilitet och prestanda för Miami.

1.10 Miami.guide/NODE_MENUS

Menyer

Beskrivning av alla menyalternativ:

Arkiv/Om...

Visa information om Miami.

Arkiv/Om MUI...

Visa information om MUI (Magic User Interface). Detta menyalternativ är endast tillgängligt om du använder någon av MUI-motorerna.

Arkiv/Ikonifiera

Ikonifiera alla Miamis fönster. Observera att detta alternativ är identiskt med "Arkiv/Stäng GUI" i vissa gränssnittsmotorer, till exempel GTLayout.

Arkiv/Stäng GUI

Ikonifiera alla Miamis fönster och ta bort GUI-motorn från minnet.

Arkiv/Koppla ned utan att lägga på

Koppla ned från servern utan att lägga på luren.

Arkiv/Avsluta utan att lägga på...

Avsluta Miami utan att lägga på luren.

Arkiv/Avsluta...

Avsluta Miami.

Inställningar/Ladda in...

Ladda in en inställningsfil.

Inställningar/Spara

Spara de nuvarande inställningarna till den aktuella inställningsfilen.

Inställningar/Spara som...

Spara de nuvarande inställningarna till en ny inställningsfil.

Inställningar/Spara som förvalda

Spara de nuvarande inställningarna som Miamis förvalda inställningar.

Inställningar/Skapa ikoner?

Skapa en ikon för varje sparad inställningsfil.

Inställningar/Importera från distribution...

Importerera inställningarna från en fil i Miamis distributionsformat (ASCII).

Inställningar/Exporterera för distribution...

Exporterera inställningarna till en fil i Miamis distributionsformat (ASCII).

Inställningar/Importerera från MiamiInit v2...

Importerera en inställningsfil från MiamiInit version 2. Denna funktion är föråldrad. Använd istället MiamiInit version 3 och importeringsfunktionen "Inställningar/Importerera från MiamiInit v3...".

Inställningar/Importerera från MiamiInitSANA2 v2...

Importerera en inställningsfil från MiamiInitSANA2 version 2. Denna funktion är föråldrad. Använd istället MiamiInit version 3 och importeringsfunktionen "Inställningar/Importerera från MiamiInit v3...".

Inställningar/Importerera från MiamiInit v3...

Importerera en inställningsfil från MiamiInit version 3.

Inställningar/MUI...

Öppna MUI-inställningarna för Miami. Detta menyalternativ är endast tillgängligt om du använder någon av MUI-motorerna.

1.11 Miami.guide/NODE_CONFIGURATION

Inställningar

Konfigureringen av Miami sker helt via ett grafiskt användargränssnitt. Du behöver inte manuellt ändra i några inställningsfiler eller miljövariabler.

Beskrivning av det grafiska användargränssnittet:

Sidan Allmänt

Sidan Gränssnitt

Sidan PPP

Sidan Uppringare

Sidan Databas

Sidan TCP/IP

Sidan Händelser

Sidan Modem

Sidan Loggning
Sidan Fönster
Sidan GUI
Sidan SOCKS
Diverse GUI-medlemmar

1.12 Miami.guide/NODE_GUI_GENERAL

Allmänt

=====

Inte så mycket här, förutom den officiella Miamilogotypen och en knapp för att starta registreringsprogrammet för Miami.

Med vissa GUI-motorer, till exempel MUI, kan du välja denna sida genom att klicka på "Allmänt" i listan längst till vänster. Med andra GUI-motorer visas den här sidan hela tiden i egenskap av Miamis huvudfönster. Andra sidor visas då som underfönster till denna.

Registrera

Knappen "Registrera"

1.13 Miami.guide/NODE_GUI_GENERAL_REGISTER

Registrera

Genom att klicka på denna knapp startar du programmet MiamiRegister, som låter dig beställa en licenskod för Miami, registrera Miami eller uppgradera din registrering. Programmet MiamiRegister måste ligga i samma låda som Miami, eller i sökvägen.

1.14 Miami.guide/NODE_GUI_INTERFACE

Gränssnitt

=====

Typ	Typ av gränssnitt
Drivrutin/Enhet	Drivrutin att använda
Hastighet	Överföringshastighet
Använd CD	Uppkopplingskontroll
Protokoll	Överföringsprotokoll
Handskak.	Typ av handskakning
EOF-läge	EOF-hantering
Serieläge	Seriellt överföringsläge
MNI-argument	Argument för MNI
IP-tilldel./IP-adress	Tilldelning av IP-adress
Nätmasktyp/Nätmask	Nätmask
Bryggtyp/Brygga	Standardbrygga
Multicast	Typ av multicast
Mappning	Typ av mappning
MTU	Maximal paketstorlek
SANA-II-parametrar	Parametrar för SANA-II
MNI-parametrar	Parametrar för MNI
MNI-information	Information om MNI-drivrutin
Hitta kort	Leta efter nätverkskort

Inaktivitet

Inaktivitetskontroll

1.15 Miami.guide/NODE_GUI_INTERFACE_TYPE

Typ

Med denna cykelknapp kan du välja vilken typ av gränssnitt du vill använda. Innehållet på sidan "Gränssnitt" varierar beroende på vilken typ du väljer, det vill säga endast de knappar och fält som är tillämpbara på den valda gränssnittstypen visas.

Tillgängliga typer:

intern seriell (PPP/(C)SLIP)

Detta är en intern implementation av PPP eller (C)SLIP som körs ovanpå Miamis egna seriella drivrutin för den interna serieporten i Amigan. Denna gränssnittstyp kan endast användas med Amigans interna serieport och kräver inga ytterligare drivrutiner. Vanligtvis uppnås bättre prestanda med denna typ än med typen "seriell drivrutin (PPP/(C)SLIP/IP)" och drivrutinen "serial.device" eller någon annan drivrutin.

seriell drivrutin (PPP/(C)SLIP/IP)

Detta är en intern implementation av PPP eller (C)SLIP som körs ovanpå en drivrutin som är kompatibel med "serial.device". Det här gränssnittet kräver ingen SANA-II-drivrutin.

SANA-II punkt till punkt

Detta är ett SANA-II-gränssnitt för en enhet som ansluter exakt två datorer, till exempel SLIP, PPP eller PLIP.

SANA-II Ethernet

Detta är ett SANA-II-gränssnitt för ett Ethernetkort, till exempel A2065, Hydra eller Ariadne.

SANA-II "gammal" Arcnet

Detta är ett SANA-II-gränssnitt för ett Arcnetkort, till exempel A2060. Denna typ använder den "gamla" Arcnetinkapslingen enligt RFC1051, vilken är populärare bland Amiganätverk än den "nya" RFC1201-inkapslingen. Använd den "gamla" inkapslingen när du vill använda din Amiga med AmiTCP/IP, Inet-225 eller NetBSD 1.1.

SANA-II "ny" Arcnet

Detta är ett SANA-II-gränssnitt för ett Arcnetkort, till exempel A2060. Denna typ använder den "nya" RFC1201-inkapslingen. Den kan inte användas med AmiTCP/IP eller NetBSD 1.1, men du kan behöva denna "nya" standard om du vill ansluta din Amiga till andra plattformar, till exempel Windows 95.

SANA-II annan buss/ring

Detta är ett SANA-II-gränssnitt för en annan buss- eller ringanslutning än Ethernet eller Arcnet. Välj denna typ om din hårdvara kan ansluta mer än två datorer men inte är kompatibel med vare sig Ethernet eller Arcnet.

1.16 Miami.guide/NODE_GUI_INTERFACE_DEVICE

Drivrutin/Enhet

Seriella gränssnitt

Ange namnet på drivrutinen och enhetsnumret för den serieport som ditt modem är anslutet till. Om du använder den interna serieporten, ange "serial.device" och "0" eller "artser.device" och "0". Du bör för närvarande inte använda "8n1.device" eftersom drivrutinen innehåller fel. Flera användare har även rapporterat om fel i "BaudBandit.device" och "v34serial.device".

När det gäller seriekort, använd drivrutinen som följer med kortet, till exempel "gvpser.device", samt det korrekta enhetsnumret.

SANA-II-gränssnitt

Ange namnet på drivrutinen och enhetsnumret för din SANA-II-hårdvara. Enhetsnumret är oftast 0.

MNI-gränssnitt

Ange namnet på drivrutinen och enhetsnumret för din MNI-hårdvara. Om du inte vet det korrekta enhetsnumret räcker det om du anger namnet på drivrutinen. Klicka sedan på "Hitta kort", välj ditt kort i fönstret som visas och klicka slutligen på "OK". Då sätts enhetsnumret till det korrekta värdet.

1.17 Miami.guide/NODE_GUI_INTERFACE_SPEED

Hastighet

(Denna funktion är endast tillgänglig hos seriella drivrutiner)

Här anger du hastigheten för din serieport. När det gäller den interna serieporten bör du sätta hastigheten till 19200, 38400 eller, om du har en snabb processor och ett grafikkort, 57600. Vid användning av seriekort kan det ibland gå bra att sätta hastigheten till 115200 eller 230400.

Använd inte 31250. Denna hastighet är reserverad för MIDI och fungerar normalt inte med modem.

Använd heller inte 14400, 28800 eller 33600. Även om ditt modem

kanske kan ansluta sig till andra modem med dessa hastigheter kan det förmodligen inte tala med serieporten med samma hastigheter.

1.18 Miami.guide/NODE_GUI_INTERFACE_CD

Använd CD

(Denna funktion är endast tillgänglig hos seriella drivrutiner)

Om funktionen "Använd CD" är påslagen använder Miami signalen "Carrier Detect" från ditt modem för att avgöra om modemmet är uppkopplat.

Detta kan vara användbart om du startar om din Amiga utan att först lägga på luren. Efter omstarten kan Miami koppla upp sig utan att behöva ringa upp igen.

Denna funktion kan endast användas om du har ställt in ditt modem så att signalen "Carrier Detect" verkligen rapporterar att modemmet är uppkopplat när det är det och att det inte är uppkopplat när det inte är det.

Vissa modem har fabriksinställningar där signalen "Carrier Detect" alltid rapporterar att modemmet är uppkopplat trots att det kanske inte är det. Om detta är fallet för ditt modem måste du ändra inställningarna för ditt modem i initieringskommandot, oftast "AT&C1", och sedan spara inställningarna till NV-RAM-kretsen i modemmet; detta kan du göra via ett terminalprogram. Gör du inte det bör du slå av "Använd CD"-funktionen.

Om du använder ett nollmodem, vilket du ställer in på sidan "Modem", innebär denna kryssruta något helt annat:

- * Om rutan är ikryssad utförs inte uppringningsskriptet.
- * Om rutan inte är ikryssad utförs uppringningsskriptet, men Miami ringer inte upp, det vill säga ATDT-kommandot hoppas över. Därmed används inte listan med telefonnummer.

1.19 Miami.guide/NODE_GUI_INTERFACE_PROTOCOL

Protokoll

(Denna funktion är endast tillgänglig hos seriella drivrutiner)

Detta är det protokoll som din Internetleverantör använder. För närvarande stöds SLIP/(C)SLIP och PPP.

1.20 Miami.guide/NODE_GUI_INTERFACE_FLOW

Handskakning

(Denna funktion är endast tillgänglig hos externa seriella drivrutiner. Den interna seriella drivrutinen använder alltid RTS/CTS)

Miami stöder två typer av handskakning: hårdvaruhandskakning (RTS/CTS) och mjukvaruhandskakning (Xon/Xoff). Det förinställda värdet är hårdvaruhandskakning och du rekommenderas starkt att inte ändra detta.

Om du inte kan använda hårdvaruhandskakning, vanligtvis beroende på ett trasigt modem, bör du ändra till mjukvaruhandskakning. Kom dock ihåg att du då även måste ändra på modemets initieringskommando, på sidan "Modem", så att det avspeglar de nya förhållandena. Mjukvaruhandskakning kan endast användas med PPP, inte med SLIP/(C)SLIP.

1.21 Miami.guide/NODE_GUI_INTERFACE_EOF

EOF-läge

(Denna funktion är endast tillgänglig hos externa seriella drivrutiner. Den interna seriella drivrutinen har alltid EOF-läget påslaget)

Miami kan upptäcka slutet på inkommande paket på två sätt: Det effektivaste sättet, som utnyttjar mindre CPU-tid, använder flaggan EOF_MODE. Detta är dock endast möjligt om den seriella drivrutinen du använder stöder EOF-läget. Många tredjepartsdrivrutiner gör inte detta.

Du bör normalt välja "automatiskt" för att låta Miami använda det förinställda värdet. Om du vet helt säkert att din drivrutin stöder EOF-läget kan du välja "på" för att upphäva det förinställda värdet. Vet du helt säkert att din drivrutin inte stöder EOF-läget kan du istället välja "av".

1.22 Miami.guide/NODE_GUI_INTERFACE_SERIAL

Seriellt läge

(Denna funktion är endast tillgänglig hos externa seriella drivrutiner. Den interna seriella drivrutinen använder alltid 8N1)

Här ställer du in antalet databitar och pariteten som används

under uppringning. Nästan alla leverantörer använder 8N1. Några få, till exempel en del av Compuserves linjer, kräver dock 7E1 eller 7O1.

Observera att de här inställningarna endast gäller vid uppringning och inloggning. (C)SLIP/PPP-protokollen använder alltid 8N1, oavsett vad du anger här. Det är helt omöjligt att använda PPP eller (C)SLIP på 7-bitarslinjer - detta gäller alla implementationer och är alltså inte en begränsning hos Miami.

1.23 Miami.guide/NODE_GUI_INTERFACE_MNIOPT

MNI-argument

(Denna funktion är endast tillgänglig hos MNI-drivrutiner)

Fältet "MNI-argument" låter dig ange olika argument för att konfigurera MNI-drivrutinen. De argument du kan använda varierar från drivrutin till drivrutin. Se

MNI-drivrutiner

för närmare information

om MNI-drivrutiner och de argument som stöds.

1.24 Miami.guide/NODE_GUI_INTERFACE_IP

IP-tilldelning/IP-adress

Internetleverantörer kan erbjuda dig två olika typer av Internetuppkoppling: antingen tilldelar de din Amiga en statisk, permanent IP-adress eller också tilldelas din Amiga en ny adress varje gång du loggar in, en så kallad dynamisk adress.

Seriella gränssnitt

Om din Amiga har en fast IP-adress, välj "statisk" och ange IP-adressen som du får av din leverantör. Om din leverantör tilldelar dig en dynamisk adress vid varje inloggning, välj "dynamisk"; Miami bestämmer då automatiskt din IP-adress under uppkopplingen.

Om du använder TIA eller Slirp måste du välja "statisk" och ange den pseudo-IP-adress som TIA eller Slirp tilldelar din Amiga. Se dokumentationen för TIA/Slirp för närmare information om detta.

SANA-II "punkt till punkt"-gränssnitt

Om din dator har en fast IP-adress, välj "statisk" och ange IP-adressen. Om adressen tilldelas din dator av en lokal

BootP-/DHCP-server, välj "DHCP". Om SANA-II-drivrutinen bestämmer den dynamiska adressen själv, till exempel genom ppp.device, välj "SANA-II".

SANA-II buss-/ringgränssnitt

Om din dator har en fast IP-adress, välj "statisk" och ange IP-adressen. Om adressen tilldelas din dator av en lokal BootP-/DHCP-server, välj "DHCP". Om adressen tilldelas din dator av en lokal RARP-server, välj "RARP".

1.25 Miami.guide/NODE_GUI_INTERFACE_MASK

Nätmasktyp/Nätmask

(Denna funktion är endast tillgänglig hos SANA-II buss-/ringdrivrutiner)

Du måste ställa in din nätmask så att Miami vet hur många datorer som finns i ditt lokala nätverk. Detta kan du göra på tre olika sätt:

statisk

Fråga din nätverksleverantör om den korrekta nätmasken och ange den här.

DHCP

Miami försöker erhålla den korrekta nätmasken från en lokal BootP-/DHCP-server.

ICMP

Miami försöker erhålla den korrekta nätmasken från en lokal server som stöder nätmaskupptäckt via ICMP.

1.26 Miami.guide/NODE_GUI_INTERFACE_GWAY

Bryggtyp/Brygga

(Denna funktion är endast tillgänglig hos SANA-II buss-/ringdrivrutiner)

Du måste ange en standardbrygga så att Miami vet var det ska skicka paket som inte är ämnade för datorer i ditt lokala nätverk. Detta kan du göra på tre olika sätt:

statisk

Fråga din nätverksleverantör om den korrekta bryggan och ange den här.

DHCP

Miami försöker erhålla den korrekta bryggan från en lokal BootP-/DHCP-server.

ICMP

Miami försöker erhålla den korrekta bryggan från en lokal server som stöder bryggupptäckt via ICMP.

1.27 Miami.guide/NODE_GUI_INTERFACE_MULTICASTS

Multicast

(Denna funktion är endast tillgänglig i den registrerade versionen)

Miami stöder nivå två av multicast, det vill säga både sändning och mottagning av multicastmeddelanden.

Om du vill använda applikationer som kräver stöd för multicast - det finns inga sådana applikationer tillgängliga ännu - kanske du måste slå på multicast i Miami. De möjliga inställningarna är:

avslaget

Multicast är avslaget.

som broadcast

Multicastmeddelanden skickas som broadcast på länknivån, eller som vanliga paket hos "punkt till punkt"-drivrutiner.

som multicast

Multicastmeddelanden skickas som multicast på länknivån. Denna funktion är endast tillgänglig hos Ethernetkort.

Observera: Multicast ska endast slås på för ett gränssnitt om du tar emot din multicastström direkt från detta gränssnitt. Om du får strömmen genom en tunnel som använder MiamiRouteD bör du vanligtvis slå av multicast i Miami; MiamiRouteD hanterar nämligen multicast själv.

1.28 Miami.guide/NODE_GUI_INTERFACE_MAPPING

Mapping

(Denna funktion är endast tillgänglig hos SANA-II Arcnetdrivrutiner)

Arcnet stöder två olika standarder för att mappa IP-adresser till hårdvaruadresser:

ARP

ARP-baserad (Address Resolution Protocol) mappning. Denna standard är förinställd och rekommenderas i de flesta fall; det är också denna standard som AmiTCP/IP använder.

direkt

Direkt mappning. Denna standard innebär att de åtta minst signifikanta bitarna i varje IP-adress mappas till hårdvaruadressen. NetBSD 1.1 använder denna standard.

Om du har minst en dator som kör NetBSD 1.1 i ditt Arcnetnätverk kan du göra livet lite lättare genom att välja "direkt" mappning istället för att manuellt skapa ARP-poster på varje dator i nätverket.

I alla andra fall bör du välja "ARP" på alla datorer. NetBSD 1.2 och högre stöder ARP för Arcnet. Om du använder någon av dessa nyare NetBSD-versioner, välj ARP-baserad mappning i Miami.

1.29 Miami.guide/NODE_GUI_INTERFACE_MTU

MTU

(Denna funktion är endast tillgänglig hos seriella drivrutiner. MTU-värdet för SANA-II-drivrutiner sätts i SANA-II-parametrar)

MTU (Maximum Transfer Unit) är storleken på det största paket som kan överföras.

Rekommenderade värden:

- * för modemhastigheter upp till 19200 BPS, sätt MTU till 296
- * för modemhastigheter på 19200 BPS och uppåt, sätt MTU till 552

Observera att bara för att du ändrar MTU-värdet på denna sida behöver det inte betyda att den största paketstorleken verkligen ändras till detta värde:

(C)SLIP kan inte förhandla om MTU, det vill säga MTU-värdet som ställs in här påverkar bara storleken på utgående paket, inte storleken på de inkommande paketen.

PPP har möjlighet att förhandla om MTU. Miami försöker alltid att sätta MTU till det värde du anger här, men den andra värden kanske motsätter sig detta och tvingar fram ett annat MTU-värde. Inträffar detta måste kanske Miami använda detta andra värde i den ena eller båda riktningarna.

Observera också att MTU-värdet inte är av avgörande betydelse när det gäller PPP. Din uppkoppling kommer fortfarande att fungera om

det MTU-värde du väljer här är lägre eller högre än det optimala värdet. När det gäller (C)SLIP måste du dock se till att ditt MTU-värde inte är högre än MTU-värdet hos din Internetleverantör.

1.30 Miami.guide/NODE_GUI_INTERFACE_STP

SANA-II-parametrar

(Denna funktion är endast tillgänglig hos SANA-II-drivrutiner)

Knappen "SANA-II-parametrar..." öppnar ett fönster där du kan ange SANA-II-inställningarna på drivrutinens länknivå. Dessa inställningar är:

- * Hårdvaruadressen till ditt nätverkskort, samt en kryssruta för att ersätta denna adress med en egendefinierad adress - detta gäller dock bara buss-/ringdrivrutiner. Hårdvaruadresser består av en serie tecken på hexadecimal form, åtskilda av ":", till exempel "01:23:45:67:89:ab".
- * Länknivåpakettyperna för IP-, ARP- och RARP-paket. RARP fungerar inte hos Arcnetdrivrutiner och varken ARP eller RARP fungerar hos "punkt till punkt"-gränssnitt.
- * Nätverkskortets MTU.
- * Antalet I/O-förfrågningar per IP- och ARP-paket. ARP fungerar inte hos "punkt till punkt"-gränssnitt.

I normala fall bör du sätta alla dessa värden till de förinställda värdena, genom att klicka på "Fråga drivrutinen", vilket endast kan ske när Miami är nedkopplat. Vid behov kan du dock upphäva de förinställda värdena, till exempel om du använder en ny typ av hårdvara som Miami inte känner till de korrekta förinställda värdena för.

1.31 Miami.guide/NODE_GUI_INTERFACE_MNIP

MNI-parametrar

(Denna funktion är endast tillgänglig hos MNI-drivrutiner)

Knappen "MNI-parametrar..." öppnar ett fönster där du kan ange MNI-inställningarna på drivrutinens länknivå. Dessa inställningar är:

- * Hårdvaruadressen till ditt nätverkskort, samt en kryssruta för

att ersätta denna adress med en egendefinierad adress - detta gäller dock bara buss-/ringdrivrutiner. Hårdvaruadresser består av en serie tecken på hexadecimal form, åtskilda av ":", till exempel "01:23:45:67:89:ab".

- * Länknivåpakettyperna för IP-, ARP- och RARP-paket. RARP fungerar inte hos Arcnetdrivrutiner och varken ARP eller RARP fungerar hos "punkt till punkt"-gränssnitt.
- * Nätverkskortets MTU.
- * Antalet I/O-förfrågningar per IP- och ARP-paket. ARP fungerar inte hos "punkt till punkt"-gränssnitt.

I normala fall bör du sätta alla dessa värden till de förinställda värdena, genom att klicka på "Fråga drivrutinen", vilket endast kan ske när Miami är nedkopplat. Vid behov kan du dock upphäva de förinställda värdena, till exempel om du använder en ny typ av hårdvara som Miami inte känner till de korrekta förinställda värdena för.

1.32 Miami.guide/NODE_GUI_INTERFACE_MNIINFO

MNI-information

(Denna funktion är endast tillgänglig hos MNI-drivrutiner)

Knappen "MNI-information" öppnar ett fönster med information om den aktuella MNI-drivrutinen; informationen består av versionsnummer, upphovsrättsinformation och en lista på kort som stöds av denna drivrutin.

1.33 Miami.guide/NODE_GUI_INTERFACE_FINDB

Hitta kort

(Denna funktion är endast tillgänglig hos MNI-drivrutiner)

Knappen "Hitta kort" aktiverar den aktuella MNI-drivrutinen och letar i ditt system efter Ethernetkort som stöds av denna drivrutin. Därefter öppnas ett fönster som visar alla kort som har hittats. Välj det kort du vill använda och klicka sedan på "OK". Detta förfarande ställer automatiskt in rätt enhetsnummer.

1.34 Miami.guide/NODE_GUI_INTERFACE_INACTIVITY

Inaktivitet

Vissa Internetleverantörer lägger på luren efter en stunds inaktivitet på linjen; detta för att förhindra användare att hålla linjer som egentligen inte används upptagna.

Gruppen "Inaktivitet" låter dig ställa in Miami att simulera aktivitet på linjen trots att du kanske inte använder den; detta för att din leverantör inte ska lägga på luren.

Med cykelknappen väljer du vilken typ av aktivitet du vill simulera: PPP-ping eller ICMP-ping. PPP-ping kräver mindre bandbredd men fungerar bara med PPP, inte med (C)SLIP, och har dessutom ingen effekt på en del leverantörer. ICMP-ping kräver något mer bandbredd men fungerar med både PPP och (C)SLIP och bör ha effekt på alla leverantörer.

Välj ICMP-ping om du använder (C)SLIP. Om du använder PPP, pröva först PPP-ping och om din leverantör ändå lägger på luren, välj då ICMP-ping.

I fältet kan du ange hur många minuter som ska passera mellan varje ping. Du måste pröva dig fram till ett värde som fungerar för dig. Vanliga värden är 9 respektive 14, vilka förhindrar att leverantören lägger på luren efter 10 respektive 15 minuter.

Observera att du först måste kontrollera om din Internetleverantör tillåter att du simulerar aktivitet på det här sättet. En del leverantörer tillåter av princip inte sådana här simulatorer och om du då använder en bryter du mot deras regler. Miamis skapare är inte ansvarig för eller skyldig till de konsekvenser som kan resultera från felaktig användning av denna aktivitetssimulator.

Observera också att det kan finna flera skäl till att ditt modem kopplas ned. Förutom inaktivitetsprincipen, som alltså kan förhindras av den här funktionen, kan det bero på en dålig linje. Det senare går inte att förhindra med hjälp av mjukvara.

Den här funktionen låter dig förhindra nedkoppling på grund av inaktivitet. En del användare vill det motsatta: koppla ned automatiskt efter en stunds inaktivitet, för att spara telefonkostnaderna. Programmet

MiamiRemind
hjälper dig med detta.

1.35 Miami.guide/NODE_GUI_PPP

PPP

===

PAP/CHAP	Autentisering
Motringning	Motringning
VJC	Komprimera ↔ pakethuvuden
ACCM	Omvandla ↔ kontrolltecken
Snabb återuppkoppling	Snabb uppkoppling efter omstart
Escape	Omvandla vanliga tecken
Erhåll DNS genom IPCP	Leta efter DNS-serverar
TermReq innan nedkoppling	Alternativ nedkopplingsmetod

1.36 Miami.guide/NODE_GUI_PPP_CHAP

PAP/CHAP

PAP och CHAP är olika protokoll som PPP använder för att skicka användarnamn och lösenord till PPP-servern.

För det mesta är användarnamnet och lösenordet desamma som dem du använder i uppringningsskriptet. Om detta är fallet för dig, kryssa för "Som i uppringing."

Om din leverantör kräver att PAP-/CHAP-användarnamnet och -lösenordet är annorlunda än dem du använder i uppringaren, kryssa inte för "Som i uppringing.", utan ange manuellt vilket användarnamn och lösenord du vill använda för PAP/CHAP.

Registrerade användare som har installerat MiamiSSL 1.2 eller högre kan kryssa för "Tillåt MS-CHAP". Detta förbättrar kompatibiliteten med vissa felinställda PPP-serverar under Windows NT. Om du har kryssat för denna ruta faller Miami tillbaka till användande av PAP när servern ber om MS-CHAP.

1.37 Miami.guide/NODE_GUI_PPP_CALLBACK

Motringning

(Denna funktion är endast tillgänglig i den registrerade versionen)

PPP stöder motringning enligt CBCP-protokollet. Om din leverantör är inställd på detta kan du be honom ringa tillbaka till dig för att du ska kunna spara telefonkostnader.

Beroende på konfigurationen hos din leverantör kan du antingen välja "Fast CBCP" eller "Dyn. CBCP". I det första fallet ringer leverantören tillbaka till ett fördefinierat telefonnummer, i det andra fallet ringer han till det nummer som du anger i fältet under denna cykelknapp.

"Värdens väntetid" är den tid som du ber värden (leverantören) att vänta innan han ringer tillbaka till dig. Denna tid bör vara tillräckligt lång för att ditt modem ska hinna koppla ned och starta om.

"Miamis väntetid" är den maximala tid som du vill att Miami ska vänta på en motringning.

1.38 Miami.guide/NODE_GUI_PPP_VJC

VJC

Van Jacobsen Compression är en teknik som används för att spara bandbredd genom att komprimera huvudena på TCP-paketen. Denna funktion bör normalt vara påslagen, vilket betyder att PPP automatiskt försöker förhandla om VJC och använder denna teknik om den andra värden går med på det.

Vissa gamla, felaktiga PPP-servrar stöder inte VJC fullt ut, så du måste kanske slå av VJC vid uppkoppling mot dem.

VJC lägger sig inte i den eventuella datakompressionen hos ditt modem, det vill säga du bör inte slå av VJC bara för att ditt modem stöder MNP-5 eller V.42bis. VJC kan användas helt oberoende av MNP-5 och V.42bis.

1.39 Miami.guide/NODE_GUI_PPP_ACCM

ACCM

PPP-protokollet stöder en mängd av kontrolltecken som omvandlas

till "escapesekvenser" under överföring, det vill säga varje tecken ersätts med en sekvens om två tecken. Denna mängd tecken kallas för ACCM (Asynchronous Control Character Mask).

Anledningarna till att denna mängd existerar är dels att PPP ska vara säkrare på linjer som inte är helt 8-bitarsberoende och dels att modemets mjukvaruhandskakning inte ska störa PPP-protokollet.

Normalt omvandlas bara ASCII-tecknen 17 och 19 (Xon/Xoff) till escapesekvenser, så att PPP kan användas vid mjukvaruhandskakning. Om du kör PPP via telnet måste du kanske omvandla fler tecken till escapesekvenser. Varje tecken som du omvandlar reducerar prestandan hos PPP med 0,8 procent.

För att ändra ACCM-inställningarna kan du antingen ange 32-bitarsvärdet direkt som hexadecimala siffror eller också klicka på popuppknappen och manuellt välja vilka kontrolltecken du vill omvandla.

1.40 Miami.guide/NODE_GUI_PPP_QUICK

Snabb återuppkoppling

Normalt tillåter Miami dig att logga in hos din leverantör igen, till exempel efter en omstart av din Amiga, utan att du behöver slå numret igen; givetvis fungerar detta endast om ditt modem fortfarande är uppkopplat. Den här funktionen fungerar dock endast om du har kryssat för "Använd CD" på sidan "Gränssnitt".

Vissa leverantörer tillåter dock inte en sådan återuppkoppling, och därmed omförhandling av PPP, utan lägger istället på luren när du försöker logga in dig igen.

I sådana här fall kan "Snabb återuppkoppling" ofta vara till hjälp. Om "Snabb återuppkoppling" är påslaget, antingen genom att du väljer "RAM" eller "fil", försöker inte Miami omförhandla PPP utan hoppar över denna fas. Istället läses alla PPP-parametrar in från antingen ett område i minnet, som har ställts in att överleva en omstart, om du har valt "RAM", eller från en fil på hårddisken om du har valt "fil". Detta förfarande låter dig i de flesta fall logga in hos din leverantör efter en omstart utan att behöva ringa upp igen.

Observera dock att om du använder inställningen "fil" och din Amiga kraschar, till exempel av ett dåligt skrivet program, medan Miami skriver återuppkopplingsfilen till din hårddisk, är det möjligt att hårddisken inte kan valideras eller råkar skadas på något sätt; detta beror i så fall på fel eller andra tillkortakommanden hos AmigaOS.

Det är därför säkrare att använda inställningen "RAM" eftersom Miami då inte behöver skapa en fil på hårddisken. Denna inställning fungerar dock endast om du undviker att starta om Amigan eller om du gör en varmstart av den. Om kraschen är mycket svår

eller om du måste kallstarta din dator, försvinner PPP-parametrarna och RAM-inställningen kan inte utföra en korrekt återuppkoppling.

1.41 Miami.guide/NODE_GUI_PPP_ESCAPE

Escape

PPP kan förhandla om att omvandla tecken med ASCII-koderna 0-31 och 128-159 till escapesekvenser. Detta ställer du in i ACCM-fältet.

Det finns dock situationer där du måste omvandla andra tecken till escapesekvenser, till exempel tecknet med den hexadecimala koden 0xFF vid rloginsessioner.

Om detta är fallet för dig, ange de tvåsiffriga hexadecimala koderna för dessa tecken, åtskilda av mellanslag, i fältet "Escape". Miami kommer då att omvandla dessa tecken till escapesekvenser när det skickar PPP-paket.

Observera att detta, till skillnad från i ACCM-fallet, endast fungerar i en riktning: när du skickar data. Om datan som skickas till dig från servern hos din leverantör också kräver omvandling av olika tecken till escapesekvenser måste du ställa in PPP-servern för detta.

1.42 Miami.guide/NODE_GUI_PPP_DNSIPCP

Erhåll DNS genom IPCP

Denna ruta är ikryssad från början. Detta betyder att Miami försöker använda sig av IPCP-utökningar för automatisk upptäckt av DNS-serverar.

Tyvärr kan det hända att vissa felaktiga PPP-serverar varken stöder denna funktion eller avvisar den på rätt sätt; de följer helt enkelt inte protokollet. Om du märker att Miami har problem med att ställa in länknivåprotokollet vid uppkoppling mot din Internetleverantör måste du kanske slå av den här funktionen.

1.43 Miami.guide/NODE_GUI_PPP_TERMREQ

TermReq innan nedkoppling

Den här funktionen bör normalt vara påslagen. Är den det skickar

Miami ett "LCP-TermReq"-meddelande till din leverantör när du vill lägga på luren. Detta har ofta den effekten att leverantören lägger på först, vilket gör att ditt modem kan koppla ned sig snabbare.

Tyvärr stöder inte alla PPP-servrar "LCP-TermReq" fullt ut. Om du märker att det tar onormalt lång tid att lägga på luren, pröva att slå av den här funktionen och se om det går snabbare att koppla ned dig från din leverantör.

1.44 Miami.guide/NODE_GUI_DIALER

```

=====
                                Uppringare

                                Uppringningsskript
                                Uppringningsskriptet

                                Telefonnummer
                                Telefonnummer att ringa

                                Maximal upprepning
                                Maximalt antal uppringningsförsök

                                Repetitionspaus
                                Paus efter alla telefonnummer

                                Uppringningspaus
                                Paus mellan enstaka telefonnummer

                                Lär
                                Skapa ett ↔
                                uppringningsskript

                                Användarnamn/Lösenord
                                Autentisering under uppringning

                                Logga
                                Logga uppringningen

```

1.45 Miami.guide/NODE_GUI_DIALER_SCRIPT

```

-----
                                Uppringningsskript

```

Listan överst i gruppen "Uppringningsskript" innehåller uppringningsskriptet. Du kan ändra posterna i detta genom att klicka på dem och ändra dem i fältet nedanför listan.

Knapparna längst ned i gruppen är till för att lägga till och ta bort poster från uppringningsskriptet.

Se

Kommandospråk för uppringaren
för mer information om språket

som uppringningsskriptet ska skrivas i.

Listan har en kontextmeny som du når genom att hålla nere höger musknapp över listan. Det kommer då fram en popuppmeny där du kan importera eller exportera uppringningsskriptet från eller till en ASCII-fil.

1.46 Miami.guide/NODE_GUI_DIALER_PHONE

Telefonnummer

Gruppen "Telefonnummer" fungerar i allt väsentligt som gruppen "Uppringningsskript", men har dessutom två ytterligare knappar: "Slå på" och "Slå av". Påslagna (aktiverade) telefonnummer har symbolen ">>" framför sig. Endast de telefonnummer som är påslagna kommer att ringas upp vid en uppkoppling.

I demoversionen kan endast tre telefonnummer lagras här; den registrerade versionen har ingen sådan gräns.

1.47 Miami.guide/NODE_GUI_DIALER_MAX

Maximal upprepning

Om ingen uppkoppling kan genomföras med något av de angivna telefonnumren väntar Miami den tid som anges i

Repetitionskaus

innan

programmet försöker igen, genom att börja om med det första numret i listan. Det maximala antalet gånger som Miami börjar om på det här sättet bestäms av värdet i fältet "Maximal upprepning". Överstigs detta värde ger Miami upp och avbryter uppringandet.

1.48 Miami.guide/NODE_GUI_DIALER_DELAY

Repetitionspaus

Om ingen uppkoppling kan genomföras med något av de angivna telefonnumren väntar Miami den tid som specificeras i fältet "Repetitionspaus" och försöker sedan igen, genom att börja om med det första numret i listan.

1.49 Miami.guide/NODE_GUI_DIALER_RDELAY

Uppringningspaus

Detta värde bestämmer uppehållet mellan uppringning av varje telefonnummer i listan. Normalt bör detta värde sättas till 0, vilket betyder att Miami direkt börjar ringa nästa nummer om det första numret är upptaget.

Vissa europeiska modem kräver ett visst uppehåll mellan uppringning av varje nummer. Om du har ett sådant modem måste du sätta värdet i fältet "Uppringningspaus" till ett uppehåll som är tillräckligt långt för ditt modem.

1.50 Miami.guide/NODE_GUI_DIALER_TEACH

Lär

Knappen "Lär" startar Miamis uppringare i interaktivt läge, det vill säga utan att exekvera uppringningsskriptet. All text som användaren skriver in eller som tas emot av modemmet sparas och Miami försöker sedan skapa ett korrekt uppringningsskript utifrån dessa uppgifter.

För det mesta är det MiamiInit som skapar ett uppringningsskript, inte knappen "Lär", men om din leverantör ändrar på inloggningsproceduren kan det vara bekvämare för dig att bara skapa ett nytt uppringningsskript, genom att använda "Lär", än att köra MiamiInit igen.

1.51 Miami.guide/NODE_GUI_DIALER_NAME

Användarnamn/Lösenord

Här anger du det användarnamn och det lösenord som ska användas i

uppringningsskriptet. Om "Som i uppring." är ikryssad på sidan "PPP" används dessa värden även för PAP/CHAP.

1.52 Miami.guide/NODE_GUI_DIALER_CAPTURE

Logga

Om du kryssar för rutan "Logga" och anger ett filnamn i fältet till höger om denna kommer uppringaren att spara all data som modemmet tar emot under uppringningen till denna fil; du kan alltså på detta sätt skapa en uppringningslogg.

1.53 Miami.guide/NODE_GUI_DATABASE

Databas

=====

Sidan "Databas" är motsvarigheten till filerna i lådan db hos andra TCP/IP-stacker; det vill säga, den låter dig konfigurera de flesta TCP-inställningarna i ditt system, till exempel adresserna till DNS-servrarna, en lista på alla användare, och så vidare.

Cykelknappen ovanför listan används för att växla mellan olika delar av databasen. Varje del har sin egen lista och sina egna fält och knappar för att redigera den aktuella posten.

Genom att använda listans kontextmeny kan du importera eller exportera varje del av databasen från eller till en ASCII-fil. Detta gör att du kan fortsätta använda dina gamla db/#?-filer från AmiTCP/AS-225 med Miami.

I den registrerade versionen av Miami kan du också sortera delar av en databas, importera från eller exportera till urklippet och slå samman databasen med olika ASCII-filer.

Om du använder någon av MUI-motorerna kan du flytta om en post i databasen genom att dra ut den från listan och sedan dra in den igen och släppa den på dess nya plats. Se dokumentationen till MUI för närmare information om sortering av poster i listor på detta sätt.

Varje post i databasen kan vara aktiv eller inaktiv. Aktiva poster känns igen på att symbolen ">>" står till vänster om dem. Du kan i de flesta GUI-motorer aktivera eller avaktivera poster genom att dubbelklicka på dem. Annars kan du alltid markera en post och klicka på "Slå på" eller "Slå av".

Alla poster i databasen kan markeras som "Temporär" genom att klicka på knappen "Temporär". Effekten blir att denna post inte sparas till disk när du sparar inställningarna och att den - i vissa

fall - raderas när du kopplar upp dig igen. Detta kan vara användbart om vissa poster, till exempel adresser till dynamiska DNS-servrar, inte ska användas vid nästa uppkoppling.

Miami markerar automatiskt alla adresser till dynamiska DNS-servrar samt ditt dynamiska värddomän som temporära.

Delar av databasen:

Protokoll	Protokoll som stöds
Tjänster	Tjänster som stöds
Värdar	Kända värdar
Nätverk	Kända nätverk
Domäner	Lokala domäner
DNS-servrar	DNS-servrar att använda
InetD	Tjänster som kan startas
Användare	Användare i systemet
Grupper	Grupper i systemet
ARP	ARP-poster i systemet
SOCKS	SOCKS-servrar att använda
IP-filter	Filtrering av IP-paket

1.54 Miami.guide/NODE_GUI_DATABASE_PROTOCOLS

Protokoll

Här visas en lista över alla protokoll som stöds, bestående av

namnet på protokollet, ett ID-värde och en valfri lista med alias. Den här tabellen motsvarar filen etc/protocols eller db/protocols hos andra protokollstackar.

Det ska i princip aldrig vara nödvändigt att ändra något i den här tabellen. Du bör aldrig ta bort någon av de förinställda posterna.

1.55 Miami.guide/NODE_GUI_DATABASE_SERVICES

Tjänster

Här visas en lista över alla TCP- och UDP-tjänster som stöds, bestående av namnet på tjänsten, ett ID-värde, ett protokollnamn och en valfri lista med alias. Den här tabellen motsvarar filen etc/services eller db/services hos andra protokollstackar.

En del applikationer kan kräva ändringar i den här tabellen; oftast rör det sig om tillägg av nya poster. Du bör dock aldrig ta bort någon av de förinställda posterna.

Att ta bort en post ur den här tabellen är inte det rätta sättet att stänga av dess funktion i InetD. Om du vill stänga av en server i InetD bör du stänga av den i tabellen "InetD" eller ta bort den därifrån; ta dock inte bort den från tabellen "tjänster", eftersom du då kan råka ut för konstiga fel i andra applikationer.

1.56 Miami.guide/NODE_GUI_DATABASE_HOSTS

Värdar

Här visas en lista över alla värdar och deras IP-adresser, bestående av en IP-adress, namnet på värden och en valfri lista med alias. Den här tabellen motsvarar filen etc/hosts eller db/hosts hos andra protokollstackar.

Miami lägger automatiskt till en värd med namnet "localhost", vilket motsvarar din Amiga, till denna tabell. Du kan manuellt lägga till andra mappningar för att göra översättningen från värd till IP-adress snabbare. Du bör dock endast lägga till mappningar för värdar som står under din direkta kontroll. Lägg aldrig till mappningar för värdar någon annanstans på Internet, eftersom du då inte kommer att kunna kontakta dessa värdar om de får en ny IP-adress.

1.57 Miami.guide/NODE_GUI_DATABASE_NETWORKS

Nätverk

Här visas en lista över nätverk, bestående av namnet på nätverket, ett ID-värde och en valfri lista med alias. Den här tabellen motsvarar filen etc/networks eller db/networks hos andra protokollstackar.

Den här tabellen används knappt nuförtiden och är endast implementerad för bakåtkompatibilitet med mycket gamla program och en del diagnostikprogram.

1.58 Miami.guide/NODE_GUI_DATABASE_DOMAINS

Domäner

Här visas en lista över alla lokala domäner, bestående endast av namnet på domänen. Den här tabellen motsvarar filen etc/domains eller db/domains hos andra protokollstackar.

Den här tabellen är egentligen inte något som TCP/IP behöver, men den gör det hela lite bekvämare för användaren: tabellen låter dig förkorta värddnamnen så att du endast behöver ange namnet på datorn, utan dess domän, när du refererar till en värd.

Exempel:

Anta att en dator på det nätverk du är ansluten till heter ex1.foo.edu och du brukar koppla upp dig mot denna dator ofta. Om du lägger till foo.edu till listan över domäner kan du alltid ansluta dig till ex1.foo.edu genom att bara skriva ex1.

Observera dock att förkortning av värddnamn på det här sättet endast fungerar för namn som söks fram genom DNS-sökning, inte för namn som söks fram genom tabellen "värdar". Detta betyder att om du lägger till domänen "foo.edu", har en värd i tabellen "värdar" som heter "ex1.foo.edu" på IP-adressen 10.0.0.1 och vill komma åt denna värd genom att bara skriva "ex1", måste du först lägga till aliaset "ex1" för denna värd i tabellen "värdar"; det vill säga den aktuella posten i tabellen "värdar" ska vara "10.0.0.1 ex1.foo.edu ex1".

1.59 Miami.guide/NODE_GUI_DATABASE_DNSSERVERS

DNS-servrar

Här visas en lista över DNS-servrar, bestående endast av

IP-adressen till servern.

DNS-servrar används för att mappa logiska värddamn till deras respektive IP-adresser. Det bör alltid finnas åtminstone en DNS-server i denna tabell, gärna en som ligger nära eller hos din leverantör.

Om Miami hittar andra DNS-servrar när det kopplar upp sig läggs dessa automatiskt till i listan och markeras som temporära.

1.60 Miami.guide/NODE_GUI_DATABASE_INETD

InetD

Här visas en lista över de "demoner" som startas av den inbyggda InetD-funktionen, bestående av ett namn på tjänsten, vilket motsvarar en post i tabellen "tjänster", typen av anslutning ("dgram" eller "stream"), vänteläget ("wait", "nowait" eller "dos"), användaren (oftast "root" i AmigaOS), ett filnamn på servern, namnet på processen hos servern och en lista med argument som ska skickas till servern. Den här tabellen motsvarar filen etc/inetd.conf eller db/inetd.conf hos andra protokollstackar.

Den i Miami inbyggda InetD-funktionen stöder många inbyggda tjänster: "daytime", "time", "echo", "discard", "chargen", "finger" och "auth". "auth" är egentligen samma sak som "identd".

Demoner för andra, externa tjänster kan startas automatiskt av InetD genom att lägga till en post för denna tjänst i tabellen. Om du vill installera externa demoner, till exempel ftpd eller telnetd, bör du först titta i programmets dokumentation efter det rätta formatet för att skriva in den "InetD"-post som krävs.

Av säkerhetsskäl rekommenderas att du slår av "echo"-, "discard"- och "chargen"-tjänsterna, eftersom de kan missbrukas för att försöka överbelasta en värd (så kallade "denial-of-service"-attacker).

1.61 Miami.guide/NODE_GUI_DATABASE_USERS

Användare

Här visas en lista över användarna i systemet, bestående av ett användarnamn, ett lösenord, ett användar-ID, ett grupp-ID, vilket motsvarar en post i tabellen "grupper", användarens riktiga namn, en hemkatalog och det kommando som ska utföras för att starta ett shellfönster för telnet. Den här tabellen motsvarar filen etc/passwd eller db/passwd hos andra protokollstackar.

Du behöver normalt bara en enda post i den här tabellen, den som gäller dig själv, såvida du inte vill köra demoner som ftpd och telnetd som låter andra användare ansluta sig till din Amiga.

Lösenorden lagras i ett krypterat format och visas inte i listfönstret. I kolumnen för lösenord står det

-

om den aktuella användaren inte har något lösenord, det vill säga om han kan logga in utan lösenord.

*

om den aktuella användaren inte kan logga in.

ett centrerat "x"

om den aktuella användaren har ett giltigt lösenord.

Proceduren för att ange ett lösenord varierar från GUI-motor till GUI-motor. När det gäller MUI och en del andra motorer ska du klicka på popuppknappen "Lösenord" för att ändra lösenordet. Du får då fram ett fält där du kan skriva in det nya lösenordet. I andra motorer måste du skriva in det nya lösenordet direkt i fältet under tabellen.

Om du lämnar fältet tomt kommer användaren inte att få något lösenord, vilket visas som "-". Om du endast skriver in tecknet "*" kommer användaren inte att kunna logga in, vilket visas som "*". I alla andra fall kommer den text du skriver in att användas som lösenord, vilket visas som ett centrerat "x".

Observera att eventuella lösenord inte bevaras om du importerar den här tabellen från AmiTCP. Lösenorden för alla användare kommer att nollställas, vilket visas som "-", och du måste manuellt ange dem igen. Detta beror på att den algoritm som används för att kryptera lösenord i AmiTCP inte kan användas av Miami, på grund av juridiska omständigheter. Se

Lösenord

för mer information om detta.

1.62 Miami.guide/NODE_GUI_DATABASE_GROUPS

Grupper

Här visas en lista över grupperna i systemet, bestående av ett gruppnamn, ett grupp-ID, och en valfri lista med användare hörande till denna grupp. Den här tabellen motsvarar filen etc/group eller db/group hos andra protokollstackar.

Du behöver normalt bara en enda post i den här tabellen, den som gäller dig själv, såvida du inte vill köra demoner som ftpd och telnetd som låter andra användare ansluta sig till din Amiga.

1.63 Miami.guide/NODE_GUI_DATABASE_ARP

ARP

Här visas en lista över ARP-poster i systemet, bestående av en IP-adress och en hårdvaruadress. Hårdvaruadressen ska vara skriven på den vanliga kolon-hex-notationen, till exempel "01:23:45". Den här tabellen motsvarar filen etc/ethers eller db/ethers hos andra protokollstackar.

ARP används endast hos SANA-II-gränssnitt av buss-/ringtyp och du behöver endast lägga till en ARP-post manuellt om en av de andra datorerna i ditt nätverk inte stöder ARP.

1.64 Miami.guide/NODE_GUI_DATABASE_SOCKS

SOCKS

Här visas en lista över SOCKS-posterna i systemet, bestående av en protokolltyp, ett kommando, en lista av värdar, en lista av portar och en lista av proxier. Den här tabellen bestämmer vilken proxyserver (SOCKS-server) som ska kontaktas, om någon sådan existerar.

De flesta användare behöver inte ändra något i den här tabellen. Om du inte använder SOCKS alls kan du ignorera den helt. Även om du använder SOCKS kan du i de flesta fall låta den här tabellen vara tom och endast ställa in en SOCKS-server i

SOCKS

. Du

behöver endast göra ändringar i den här tabellen om du vill att Miami ska kontakta olika SOCKS-servrar för olika värdar eller portar, eller om du har ett komplicerat lokalt nätverk, med flera delnät, bakom en SOCKS-brandvägg.

Varje post i den här tabellen definierar ett filter för en uppkoppling och en lista över de proxyserverar som ska kontaktas om en uppkoppling stämmer överens med detta filter. Vid varje uppkoppling söks tabellen igenom från början till slut och den första post som stämmer överens används; ordningen mellan posterna i tabellen spelar alltså roll. Formatet för en post är följande:

Typ

Här definierar du vilken typ av uppkoppling som ska användas, om detta filter stämmer överens. Giltiga typer är "socks4" för en SOCKS-v4-uppkoppling, "socks5" för en SOCKS-v5-uppkoppling och "noproxy" för en direkt uppkoppling, utan SOCKS.

Kommando

Detta fält är en del av filtret och utgörs av en lista med bokstäver åtskilda med kommatecken, utan några mellanslag. Varje bokstav står för en viss typ av begäran: "c" är koppla upp; "b" är bind; "u" är UDP; "p" är ping, "t" är traceroute; "-" är annan begäran.

Värdar

Detta fält är en del av filtret och utgörs av en värdddefinition med följande möjliga värden: "värdip/mask" motsvarar de värdar som har denna IP-adress och nätmask, till exempel "1.2.3.4/255.255.0.0"; "-" motsvarar alla värdar; "n1" motsvarar "n1.0.0.0/255.0.0.0"; "n1.n2" motsvarar "n1.n2.0.0/255.255.0.0"; "n1.n2.n3" motsvarar "n1.n2.n3.0/255.255.255.0"; ".domän.namn" motsvarar alla värdar som slutar med ".domän.namn"; "en.värd.namn" motsvarar bara värden "en.värd.namn".

Portar

Detta fält är en del av filtret och utgörs av en portdefinition med följande möjliga värden: "-" motsvarar alla portar; "namn" motsvarar tjänstens namn, till exempel "ftp"; "nummer" motsvarar ett portnummer, till exempel "80"; "[100,1000]" motsvarar portarna 100 till 1000; "(100,1000)" motsvarar portarna 101 till 999; "(100,1000]" motsvarar portarna 101 till 1000.

Proxier

Här definierar du vilka proxyservrar som ska kontaktas för uppkopplingar som stämmer överens med filtret för den här posten. Det här fältet utgörs av en lista över servrar, åtskilda med kommatecken. Varje server ska utgöras av ett värddnamn eller en IP-adress. Därefter kan ett kolon och ett portnummer på proxyservern följa.

Den här tabellen används endast om "Slå på SOCKS" är ikryssad på sidan

SOCKS

. Uppkopplingar som inte stämmer överens med någon av posterna i den här tabellen kontaktar automatiskt den förvalda SOCKS-servern och -porten som definieras på sidan

SOCKS

, med SOCKS v5.

1.65 Miami.guide/NODE_GUI_DATABASE_IPFILTER

IP-filter

(Denna funktion är endast tillgänglig i den registrerade versionen)

Den här tabellen låter dig filtrera bort vissa IP-paket som anländer till det lokala nätverket samt låter dig skapa poster i systemloggen om vissa paket anländer. Användandet av ett IP-filter

låter dig implementera en mycket enkel brandvägg och du kan bli meddelad när någon försöker bryta sig in i din dator.

Tabellen består av en uppsättning regler. Alla paket som anländer kontrolleras mot dessa regler, från början till slut. Den första regeln som kan tillämpas på paketet bestämmer huruvida paketet ska avvisas och om en post i systemloggen ska skapas för detta paket. Regler längre ned i tabellen kontrolleras inte om en tidigare regel har visat sig vara tillämpbar.

Varje post i tabellen består av följande delar:

- * Ett protokoll, det vill säga "tcp", "udp" eller "*", där stjärnan betyder "alla protokoll".
- * En tjänst, det vill säga ett namn som finns i tabellen "tjänster". "*" betyder "alla portar". "\$" betyder "alla tjänsteportar", det vill säga alla portar förutom de i intervallet 1024 - 5000. Du kan också ange ett intervall av tjänster genom att separera den första och den sista tjänsten med "/"; till exempel betyder "1/80" alla tjänster mellan port 1 och port 80.
- * En IP-adress som refererar till paketets källa.
- * En nätmask som definierar ett intervall av IP-adresser.
- * Två parametrar som bestämmer en handling: du kan tillåta eller avvisa åtkomst ("y" för "ja" eller "n" för "nej") och skapa eller undertrycka skapande av en post i systemloggen ("y" för "ja" eller "n" för "nej").

Observera att det endast skapas poster i systemloggen för "tcp"-tjänster, ej för "udp"-tjänster.

Här är ett exempel på en användbar inställning av IP-filtret:

```
* * 127.0.0.1 (tom mask) y n
tcp auth *.*.*.* (tom mask) y n
* $ *.*.*.* (tom mask) y y
```

Vad som sker:

Den första raden säkerställer att paket som skickas lokalt, det vill säga från din Amiga till dig själv, tillåts utan loggning.

Den andra raden tillåter också inkommande "auth"-förfrågningar utan loggning. Detta är användbart eftersom "auth"-förfrågningar ("ident"-förfrågningar) skickas av så många httpd-, ftpd- och ircd-serverar att du förmodligen inte vill ha en post i systemloggen för varje förfrågan.

Den tredje raden tillåter alla andra externa förfrågningar, men skapar en post i systemloggen som talar om för dig att någon försöker komma åt din dator. Det är viktigt att tjänsten anges som "\$" och inte som "*". Detta beror på att ftp använder sig av omvända uppkopplingar, från servern till klienten, under uppladdning

och nedladdning. Om du anger tjänsten som "*" skulle en post i systemloggen skapas varje gång du laddar upp eller laddar ned en fil till eller från en ftp-server.

Alla andra paket, det vill säga paket som skickas utifrån till en port mellan numren 1024 och 5000, använder sig av den inbyggda regeln att tillåta paketet men inte skapa en post i systemloggen.

1.66 Miami.guide/NODE_GUI_TCPIP

```

=====
TCP/IP

Värddnamn
Systemets värddnamn

Namn/Användarnamn
Användarinformation

ICMP
Verifiera via ICMP

DHCP
Dynamiska IP-nummer ↔
via DHCP

Verifiera DNS-servrarna
Verifiera alla DNS-servrar

Falsk IP
"Falsk" lokal IP-adress

T/TCP
Aktivera stöd för T/ ↔
TCP

Lägg till domänen
Lägg till värddnamnets domän

Logga ut vid nedkoppling
Stäng TCP-uppkopplingar

Skydd mot pingfloder
Skydda dig mot pingfloder

Ställ klockan
Ställ klockan via nätet

```

1.67 Miami.guide/NODE_GUI_TCPIP_HOSTNAME

Värddamn

Normalt bör du kryssa i rutan "dynamiskt". Gör du detta bestämmer Miami automatiskt din Amigas värddamn genom omvänd DNS-sökning när du kopplar upp dig.

Dessvärre stöder inte alla leverantörer omvänd DNS-sökning, eller också tilldelar de sina användare statiska värddamn som inte finns med i DNS-tabellen. Du ska i dessa fall ej kryssa i "dynamiskt" utan skriva in ditt värddamn själv.

1.68 Miami.guide/NODE_GUI_TCPIP_NAME

Namn/Användarnamn

I dessa fält ska du ange ditt riktiga namn, till exempel "Sven Svensson", och ditt användarnamn på din Amiga, till exempel "ssvensson".

Även om du teoretiskt sett kan skriva in vilka namn som helst, är det god sed att använda "riktiga" namn, inte några fiktiva namn.

Vissa program letar upp information om dig som användare genom att läsa av ditt användarnamn. För att dessa program ska uppföra sig korrekt bör du försäkra dig om att det finns en post i listan "användare" på sidan "Databas" som motsvarar det användarnamn du anger här.

1.69 Miami.guide/NODE_GUI_TCPIP_ICMP

ICMP

Om denna ruta är ikryssad använder Miami "ICMP-pingar" för att kontrollera att IP-adresser, DNS-serverar, med mera är korrekta.

Denna ruta bör normalt vara ikryssad eftersom funktionen ger dig skydd mot felaktiga inställningar.

Om du kopplar upp dig via en TCP-emulator som till exempel TIA kan det hända att du måste slå av den här funktionen, eftersom inte alla TCP-emulatorer stöder ICMP.

1.70 Miami.guide/NODE_GUI_TCPIP_BOOTP

DHCP

Om din leverantör använder sig av dynamiska IP-adresser har Miami olika tekniker för att hitta den korrekta, dynamiska IP-adressen.

Vid PPP-uppkopplingar sköts detta för det mesta av PPP-protokollet självt. (C)SLIP har dock ingen sådan funktion utan måste oftast använda sig av protokollet "DHCP", eller dess föregångare "BootP". Alternativt kan IP-adressen ibland bestämmas utifrån uppringningsloggen.

Om du använde dig av MiamiInit för att ställa in din uppkoppling kan du lämna den här rutan i sitt förinställda tillstånd. Om du konfigurerade Miami manuellt bör du först kryssa för "DHCP". Går detta bra kan du slå av funktionen och se om det fortfarande fungerar.

Om Miami kan hitta dina IP-adresser utan användning av DHCP bör du undvika att kryssa för "DHCP"; det går nämligen ofta snabbare att koppla upp sig om den här funktionen inte används.

1.71 Miami.guide/NODE_GUI_TCPIP_VERIFYDNS

Verifiera DNS-servrarna

Miami försöker normalt verifiera att IP-adresserna till alla DNS-servrarna är korrekta. Detta kan dock orsaka problem med en del Internetleverantörer som erbjuder en dålig uppkoppling eller inte svarar på förfrågningar omedelbart efter det att en uppkoppling har skett.

Om du slår av funktionen "Verifiera DNS-servrarna" kommer Miami att hoppa över det här verifieringssteget när det kopplar upp sig.

1.72 Miami.guide/NODE_GUI_TCPIP_FAKEIP

Falsk IP

Om du kopplar upp dig mot Internet via en TCP-emulator som till exempel TIA eller Slirp, och denna emulator inte tilldelar dig en "riktig" IP-adress, utan en falsk sådan, måste du kryssa för den här rutan.

Den talar om för Miami att det ska leta rätt på ditt värdnamn genom att undersöka den andra värdens IP-adress, inte din lokala, "falska" adress.

1.73 Miami.guide/NODE_GUI_TCPIP_TTCP

T/TCP

(Denna funktion är endast tillgänglig i den registrerade versionen)

T/TCP (TCP vid överföringar) är en utökad variant av TCP som kan öka hastigheten markant hos vissa applikationer, speciellt webbläsare; detta kräver dock att både webbläsaren och servern stöder T/TCP.

Registrerade användare bör normalt slå på den här funktionen för att utnyttja hastighetsökningen. Dessvärre har vissa PPP-serverar problem med att hantera utökade TCP-paket som genererats av T/TCP, så om Miami slutar fungera efter det att du har slagit på T/TCP, bör du slå av funktionen - eller byta leverantör.

1.74 Miami.guide/NODE_GUI_TCPIP_ADDDOMAIN

Lägg till domänen

Om den här rutan är ikryssad kommer Miami automatiskt att lägga till ditt värdomäns domän, det vill säga allting efter den första punkten, till Miamis domändatabas.

Detta är inte något som Miami eller några andra program kräver, men det kan vara bekvämt för dig om du vill använda förkortade värdomän. Se sidan

Databas

för mer information om vad som menas med domändatabasen.

1.75 Miami.guide/NODE_GUI_TCPIP_DOWN

Logga ut vid nedkoppling

(Denna funktion är endast tillgänglig i den registrerade versionen)

I den oregistrerade versionen av Miami kopplar programmet alltid ned alla aktiva TCP-sessioner när gränssnittet kopplar ned. I den registrerade versionen håller Miami TCP-sessionerna vid liv i en sådan situation. Detta har den fördelen att du kan koppla upp dig igen

mycket snabbt och fortsätta använda din TCP-session.

Nackdelen med att hålla TCP-sessioner vid liv när gränssnittet kopplas ned är att andra applikationer inte kan få reda om Miami är uppkopplat eller nedkopplat. Deras uppkopplingsförsök misslyckas om Miami är nedkopplat men det ges inga ytterligare felmeddelanden.

Om du inte vill att Miami ska bete sig på det här sättet, utan att det ska visa de relevanta felmeddelandena när gränssnittet är nedkopplat, bör du slå på den här funktionen.

1.76 Miami.guide/NODE_GUI_TCPIP_PING

Skydd mot pingfloder

(Denna funktion är endast tillgänglig i den registrerade versionen)

Miami använder sig av en enkel algoritm för att reducera effekterna av försök att överbelasta värdar ("denial-of-service"-attacker), som kan orsakas av pingfloder.

Om den här funktionen är påslagen och en användare försöker överbelasta din dator med pingar, genom att antingen skicka dig mycket stora pingpaket eller mindre pingpaket väldigt snabbt, kommer Miami att skapa en post i systemloggen för att meddela dig om attacken samt sluta svara på pingar från den användaren på ett tag. En tid efter det att attacken har upphört kommer Miami åter att börja svara på pingar från denna användare.

Observera att det inte finns någon möjlighet för dig att hindra användaren från att dränka dig. Den enda möjligheten att förhindra slöseri med ditt modems bandbredd är att installera ett filter hos din Internetleverantör. Allt Miami kan göra för att förhindra pingfloder är att sluta svara på dem samt att meddela dig om läget, så att du kan koppla upp dig mot en annan modempool.

1.77 Miami.guide/NODE_GUI_TCPIP_GETTIME

Ställ klockan

Om din Amiga inte är utrustad med en klocka som går på batteri bör du slå på funktionen "Ställ klockan" och i fältet ange namnet på eller IP-adressen till en server som stöder tjänsten "time". Om du är osäker på vad du ska skriva, prova vilken "viktig" dator som helst hos din leverantör, till exempel den du använder för e-post eller nyhetsgrupper.

Om du använder den här funktionen måste du försäkra dig om att

miljövariabeln "ENV:TZ" är korrekt inställd, till exempel på "EST5" eller "EST4EDT" under sommartid. Detta är viktigt eftersom servern skickar dig tiden i GMT-format (UTC-format) och Miami måste översätta denna tid till din lokala tidszon. Se
Tidszonsinformation
för mer information.

Observera: Använd inte denna funktion om din Amiga har en klocka som går på batteri, eftersom klockan i din Amiga då kan ställas bakåt. Detta kan få program som använder funktionen GetSysTime() för beräkningar att bete sig underligt och kanske krascha.

1.78 Miami.guide/NODE_GUI_EVENTS

Händelser
=====

Miami låter dig kontrollera vad som ska hända när vissa händelser inträffar. Till exempel kan Miami starta ett ARexx-program och ikonifiera sitt fönster när uppkopplingen är klar.

De olika som händelser Miami låter dig kontrollera på detta sätt är:

Start

Programmet startas.

Avlutande

Programmet avslutas.

Aktiv nedkoppling

Nedkoppling som orsakades av användaren, till exempel genom att använda knappen "Koppla ned" eller via ARexx-kommandot "OFFLINE".

Passiv nedkoppling

Nedkoppling som orsakades av att modemmet eller Internetleverantören lade på luren.

Uppkoppling

Vid uppkoppling; det vill säga uppkopplingen mot Internetleverantören är klar och alla nödvändiga protokoll har initierats.

Misslyckad uppkoppling

Ett försök att koppla upp har misslyckats av någon anledning, till exempel att alla telefonlinjer var upptagna och det maximala antalet uppringningsförsök har gjorts.

Miami kan reagera på följande sätt. Alla sätt är inte meningsfulla för alla händelser, varför bara en del kan vara tillgängliga i varje enskilt fall.

ARexx

Starta ett ARexx-program.

Shell

Starta ett Shellskript för AmigaDOS.

Göm GUI

Ikonifiera Miamis fönster.

Stäng GUI

Ikonifiera Miamis fönster och stäng GUI-motorn.

Autouppkoppling

Försök koppla upp (ringa upp) automatiskt.

Blänk

Blänk skärmen eller pip, beroende på systemets inställningar.

Behåll GUI

Avikonifiera Miamis fönster.

I utvärderingsversionen av Miami är alternativen "ARexx" och "Shell" inte tillgängliga samt "Autouppkoppling" är inte tillgängligt för händelsen "Passiv nedkoppling".

Textfältet "Konsollfönsternamn" låter dig ange den in- och utdatafil som ARexx-program och Shellskript ska använda. Det bör vara något i stil med "Con:1/1/400/100/Titel"

1.79 Miami.guide/NODE_GUI_MODEM

Modem

=====

Initieringskommando

Kommando innan uppringning börjar

Avslutningskommando

Kommando när Miami avslutas

Uppringningsprefix

Text före telefonnummer

Uppringningssuffix

Text efter telefonnummer

Nollmodem

Ett nollmodem används

1.80 Miami.guide/NODE_GUI_MODEM_INIT

Initieringskommando

Initieringskommandot för ditt modem. Sätts normalt av MiamiInit.

1.81 Miami.guide/NODE_GUI_MODEM_EXIT

Avslutningskommando

Det kommando som skickas till modemmet när Miami avslutas. Normalt behöver inget anges här, men det kan vara användbart om flera program delar på modemmet och modemmet behöver återställas till de förvalda inställningarna.

1.82 Miami.guide/NODE_GUI_MODEM_PREFIX

Uppringningsprefix

Det kommando modemmet använder för att ringa; det vill säga, den text som skickas före telefonnumret. Detta är oftast "ATDT" eller "ATDP" (eller helt enkelt "ATD").

1.83 Miami.guide/NODE_GUI_MODEM_SUFFIX

Uppringningssuffix

Den text som behöver läggas till efter telefonnumret för att avsluta uppringningskommandot. Detta är normalt "\r".

1.84 Miami.guide/NODE_GUI_MODEM_NULLMODEM

Nollmodem

Miami förväntar sig normalt att du har ett modem kopplat till din seriella port. Om din Amiga är direkt kopplad till en annan dator med en nollmodemskabel behöver du kryssa för den här rutan. Miami kommer då inte att skicka några kommandon (så kallade AT-kommandon) till

modemet eller vänta på svar som "OK" eller "CONNECT".

Om rutan "Nollmodem" är kryssad ändras betydelsen av rutan "Använd CD" på sidan "Gränssnitt":

- * Om din dator är kopplad till en dator som kräver en loginsekvens för att etablera SLIP/PPP-länken bör du inte kryssa för rutan "Använd CD". Miami använder då uppringningsskriptet som anges i uppringningsfönstret men utan att ringa ett nummer först. Det här alternativet är mycket användbart för att koppla upp mot en Unix- eller Linuxdator som kör "getty" med kontroll av användarnamn och lösenord på sin seriella port.
- * Om din dator är kopplad till en dator som använder sin seriella port uteslutande i SLIP/PPP-läge (till exempel en annan Amiga som kör Miami) bör du kryssa för rutan "Använd CD". Miami kommer då inte att använda uppringningsskriptet alls utan börjar istället direkt med protokollförhandlingen.

1.85 Miami.guide/NODE_GUI_LOGGING

```

=====
                                Logging

                                Konsoll
                                Konsollfönster för logg

                                Fil
                                Fil att logga till

                                Använd syslog.library
                                Använd syslog.library

                                Telefonlogg
                                Logga upp- och nedkopplingar

                                PPP-logg
                                Logga PPP-uppkoppling

```

1.86 Miami.guide/NODE_GUI_LOGGING_CONSOLE

Konsoll

I det här fältet kan du ange ett AmigaDOS-filnamn för det konsollfönster som Miami använder för systemets loggmeddelanden. Filen hålls öppen efter det första systemmeddelandet har visats, så

du bör använda "Con:"-modifierarna "/AUTO/CLOSE" för att du ska kunna stänga fönstret utan att tappa gamla systemmeddelanden.

1.87 Miami.guide/NODE_GUI_LOGGING_FILE

Fil

I det här fältet kan du ange namnet på den AmigaDOS-fil där Miami lagrar systemets loggmeddelanden. Om filen redan finns kommer Miami att lägga till nya meddelanden på slutet av filen; filens gamla innehåll kommer alltså inte att raderas.

1.88 Miami.guide/NODE_GUI_LOGGING_SYSLOG

Använd syslog.library

Om den här rutan är kryssad kommer Miami att försöka använda syslog.library för sin systemlog. syslog.library är en del av paketet SysLog av Petri Nordlund.

1.89 Miami.guide/NODE_GUI_LOGGING_PHONE

Telefonlogg

Miami kan logga alla upp- och nedkopplingar för att göra det lättare för dig att ha kontroll på telefonräkningen.

De två knapparna låter dig aktivera telefonloggning och ange namnet på den fil till vilken Miami lägger till räkningsposter.

För tillfället stöds bara textformat, med rader som följer:

```
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
```

Raden "Online" innehåller telefonnumret som ringdes i "()". "Reconnect" inträffar när Miami kopplar upp utan att ringa först, till exempel efter du har startat om din Amiga.

Skillnaden mellan passiv nedkoppling ("passive offline") och aktiv nedkoppling ("active offline") är att en aktiv nedkoppling är frivillig, till exempel som en följd av ARexx-kommandot "OFFLINE"

eller någon använde knappen "Koppla ned". Passiv nedkoppling är resultatet av att modemmet lade på luren eller din Internetleverantör kopplade ned dig.

1.90 Miami.guide/NODE_GUI_LOGGING_PPP

PPP-logg

(Den här funktionen är bara tillgänglig i den registrerade versionen.)

Fältet "Fil" låter dig ange namnet på en fil där Miami loggar etableringsfasen av PPP-uppkopplingen. Data loggas i en för människor läsbar form, alltså inte som en hexadecimal dump. Bara själva etableringsfasen av uppkopplingen loggas, det vill säga till och med LCP och IPCP har statusen "öppen". Efter det loggas inget mer.

Det främsta syftet med PPP-loggen är att göra det lättare att hitta kompatibilitetsproblem på PPP-nivån och för att göra det lättare att optimera PPP-alternativen för en viss PPP-server.

1.91 Miami.guide/NODE_GUI_WINDOWS

=====
 =====

Windows

Bekräfta avslutande

Bekräfta vid avslutande

Bekräfta nedkoppling

Bekräfta vid nedkoppling

Bekräfta protokollfel

Bekräfta vid protokollfel

Uppringare

Uppringarfönstrets ←
 innehåll

1.92 Miami.guide/NODE_GUI_WINDOWS_REQQUIT

Bekräfta avslutande

Du kan ange när Miami ska fråga innan programmet avslutas:

- * Alltid.
- * När program som använder Miami fortfarande körs.
- * När Miami är uppkopplat.

Eller kombinationer av dessa.

1.93 Miami.guide/NODE_GUI_WINDOWS_REQOFFLINE

Bekräfta nedkoppling

Om du kryssar för den här rutan frågar Miami innan nedkoppling sker.

1.94 Miami.guide/NODE_GUI_WINDOWS_REQERRORS

Felfönster

Normalt visar Miami ett felfönster om något problem uppstår under uppringningen eller när länken konfigureras. Om den här rutan inte är ikryssad kommer sådana fel att tyst ignoreras och Miami visar inte något felfönster.

1.95 Miami.guide/NODE_GUI_WINDOWS_DIALER

Uppringare

Det normala uppringningsfönstret består av tre delar: en hjälptext överst, ett antal knappar i mitten och ett dialogområde längst ned. Med de tre "Visa"-knapparna kan du ange vilka av dessa tre delar som ska visas.

Om ingen ruta är ikryssad kommer uppringaren att visa en enda textrad som innehåller det uppringningskommando som för tillfället utförs.

Kryssrutan "Aktivera fönster" avgör om uppringningsfönster och

felfönster ska vara aktiva när de öppnas.

1.96 Miami.guide/NODE_GUI_GUI

GUI

===

Den här sidan innehåller olika inställningar för Miamis användargränssnitt, det vill säga direktval, ikonifiering, ikoner och vilket användargränssnitt (vilken GUI-motor) som ska användas.

Viktigt: Använd alltid de här inställningarna för användargränssnittet, inte de i andra program. Även om du använder MUI, använd inte inställningarna för ikonifiering och direktval i MUI-inställningarna för Miami. De inställningarna fungerar inte med Miami eftersom Miami tar själv hand om ikonifiering utan att använda MUI.

Direktval

Direktval för ikonifiering

Visa ikon

Visa ikon när ikonifierad

Visa i meny

Visa i meny när ikonifierad

Inget GUI vid start

Öppna inget fönster vid start

Uppkopplad ikon

Ikon när Miami är uppkopplat

Nedkopplad ikon

Ikon när Miami är nedkopplat

GUI-motor

Välj GUI-motor att använda

Byt

Byt till vald GUI-motor

1.97 Miami.guide/NODE_GUI_GUI_HOTKEY

Direktval

Det här inmatningsfältet anger det Commoditiesdirektval som ikonifierar eller avikonifierar Miamis användargränssnitt. Den vanliga Commoditiessyntaxen används för att definiera direktvalet. Till exempel anger "ctrl alt m" att direktvalet är tangenten "m" i kombination med tangenten "Ctrl" och en av "Alt"-tangenterna. "ctrl alt m" är även det förvalda värdet.

1.98 Miami.guide/NODE_GUI_GUI_SHOWICON

Visa ikon

Om den här rutan är ikryssad kommer en applikationsikon att visas på Workbenchskärmen när Miami är ikonifierat.

1.99 Miami.guide/NODE_GUI_GUI_SHOWMENU

Visa i meny

Om den här rutan är ikryssad kommer det att finnas ett menyalternativ kallat "Miami" i Workbenchmenyn "Verktyg" när Miami är ikonifierat.

1.100 Miami.guide/NODE_GUI_GUI_ONSTARTUP

Inget GUI vid uppstart

Om den här rutan är ikryssad kommer Miami inte att ladda någon GUI-motor vid uppstart och inget fönster kommer att öppnas. Det här är mycket användbart i kombination med automatisk uppkoppling vid start. Se

Händelser

för mer information om detta.

1.101 Miami.guide/NODE_GUI_GUI_ONLINEICON

Uppkopplad ikon

Det här inmatningsfältet låter dig ange den ikon (".info"-fil)

som Miami använder som applikationsikon när Miami är uppkopplat. Det förvalda (när fältet är tomt) är att använda en inbyggd ikon.

1.102 Miami.guide/NODE_GUI_GUI_OFFLINEICON

Nedkopplad ikon

Det här inmatningsfältet låter dig ange den ikon (".info"-fil) som Miami använder som applikationsikon är Miami när nedkopplat. Det förvalda (när fältet är tomt) är att använda en inbyggd ikon.

1.103 Miami.guide/NODE_GUI_GUI_GUI

GUI-motor

Den här knappen låter dig välja en av flera installerade GUI-motorer. Miami kommer ihåg vad du väljer och lagrar valet i inställningsfilen (om du sparar inställningarna efteråt), men Miami byter inte omedelbart till den nya GUI-motorn. För att göra det, använd knappen

Byt

.

1.104 Miami.guide/NODE_GUI_GUI_SWITCH

Byt

Om du använder den här knappen kommer Miami att byta till den valda GUI-motorn. (Vad som verkligen händer är: Miami ikonifieras, den aktuella GUI-motorn tas bort från minnet, den nya GUI-motorn laddas och sedan avikonifieras Miami med den nya GUI-motorn.)

1.105 Miami.guide/NODE_GUI SOCKS

SOCKS

=====

Den här sidan låter dig konfigurera SOCKS-klientstödet i Miami. Om du aldrig har hört talats om SOCKS behöver du det förmodligen

inte. SOCKS är ett proxysystem för att låta datorer innanför en brandvägg kontakta datorer utanför brandväggen.

Miamis implementation av SOCKS låter Amiga-TCP/IP-klienter ta kontakt "genom" brandväggar transparent, utan speciellt stöd i klienterna. Om din nätverksleverantör använder en SOCKS-brandvägg, fråga efter SOCKS-serverns IP-adress samt användarnamn och lösenord för SOCKS (om SOCKS-servern är lösenordsskyddad) och konfigurera Miami på den här sidan.

Inställningarna på den här sidan är de förvalda inställningarna för din konfiguration. Du kan konfigurera SOCKS i mer detalj på sidan

Databas/SOCKS

.

Slå på SOCKS

Aktivera SOCKS

Förvald SOCKS-server

SOCKS-server som normalt används

Loggningsnivå

Mängd loggmeddelanden att skapa

SOCKS-autentisering

Autentisering på SOCKS-servern

1.106 Miami.guide/NODE_GUI_SOCKS_ENABLE

Slå på SOCKS

Om den här rutan är ikryssad kommer Miami använda SOCKS för att koppla upp mot alla datorer som inte kan nå direkt genom ett gränssnitt. Du behöver också konfigurera SOCKS-serverns IP-adress och portnummer, samt med vissa SOCKS-serverar, autentiseringsinformation.

1.107 Miami.guide/NODE_GUI_SOCKS_SERVER

Förvald SOCKS-server

De här inmatningsfälten anger IP-adressen och portnumret för den förvalda SOCKS-servern i ditt nätverk. Portnumret för SOCKS är oftast 1080.

1.108 Miami.guide/NODE_GUI SOCKS_MAXLOG

Loggningsnivå

Den här knappen anger hur många diagnostiska meddelanden du vill ha från SOCKS. Du bör normalt ha denna knapp på "none" (inga) eller "error" (fel). Högre värden är användbara för att få ytterligare diagnostiska meddelanden vid felsökning.

1.109 Miami.guide/NODE_GUI SOCKS_AUTH

SOCKS-autentisering

De här knapparna anger den autentiseringsinformation som skickas till SOCKS-servern. Följande autentiseringsmetoder kan användas:

Ingen

Ingen autentiseringsinformation skickas. Detta fungerar bara med SOCKS-serverar som inte kräver någon autentisering.

Som i uppringaren

Miami skickar det användarnamn och lösenord som anges i uppringaren.

Användarnamn/lösenord

Miami skickar det användarnamn och lösenord som anges i fälten nedanför cykelknappen.

1.110 Miami.guide/NODE_GUI_MISC

Diverse

=====

Det finns ytterligare tre knappar i Miami som inte beskrivs i något av de tidigare avsnitten:

- * "Koppla upp": Gör att Miami börjar ringa upp och försöker koppla upp.
- * "Koppla ned": Gör att Miami lägger på och kopplar ned.
- * En lista eller uppsättning knappar till vänster i Miami-fönstret som används för att välja en av inställningssidorna.

1.111 Miami.guide/NODE_MNI

MNI-Ethernetdrivrutiner

MNI-drivrutiner är ett nytt sätt att använda ditt Ethernetkort. Jämfört med vanliga SANA-II-drivrutiner ger MNI ofta bättre prestanda (ofta mycket bättre prestanda), ytterliggare funktioner, till exempel stöd för promiskuöst läge i MiamiTCPDump och enklare konfiguration. Kompatibiliteten med några hubbar och kabelmodem är också högre än med SANA-II i vissa fall.

Nackdelen med att använda MNI är att MNI inte stöds av andra protokollstackar ännu (till exempel Envoy), så du kan inte köra andra stackar parallellt med Miami när du använder MNI. Det kommer dock att ändras i framtiden.

För att använda ditt Ethernetkort med MNI, sätt gränssnittstypen till "MNI-Ethernet" och ange namnet på MNI-drivrutinen för ditt kort (se lista nedan). Klicka sedan på "Hitta kort" för att bekräfta att kortet stöds. Välj kortet och klicka på "OK". Detta sätter enhetsnumret korrekt. Klicka nu på "MNI-parametrar..." och "Fråga drivrutinen", sedan "OK" för att konfigurera länknivåinställningarna. Konfigurera sedan de återstående inställningarna (till exempel IP-adresser). För några MNI-drivrutiner behöver du även ändra argument i "MNI-argument". Se beskrivningen för din MNI-drivrutin för att se om det är nödvändigt.

Här är en lista över alla Ethernetkort till Amiga som stöds för närvarande, tillsammans med hänvisningar till deras respektive MNI-drivrutiner.

ASDG LanRover EB920	ASDG LanRover EB920: z2-dp8390.mni
Ariadne	Ariadne: z2-am7990.mni
Ariadne-II	Ariadne-II: z2-dp8390.mni
CEI/Ameristar A2065	CEI/Ameristar A2065: z2-am7990.mni
CEI/Ameristar A4066	CEI/Ameristar A4066: z2-smc91c90.mni
Commodore A2065	Commodore A2065: z2-am7990.mni
ConneXion	ConneXion: z2-am7990.mni
GG2-Bus+, NE2000	

```

GG2-Bus+, NE2000: gg2-dp8390.mni

Hydra AmigaNet Z2
Hydra AmigaNet Z2: z2-dp8390.mni

QuickNet QN2000
QuickNet QN2000: z2-mb86950.mni

```

Här är en lista över alla tillgängliga MNI-drivrutiner.

```

gg2-dp8390.mni
gg2-dp8390.mni

z2-am7990.mni
z2-am7990.mni

z2-dp8390.mni
z2-dp8390.mni

z2-mb86950.mni
z2-mb86950.mni

z2-smc91c90.mni
z2-smc91c90.mni

```

1.112 Miami.guide/NODE_MNI_GGTDPETNZ

```

gg2-dp8390.mni
=====

```

Drivrutin för NE2000-kompatibla ISA-kort (baserat på kretsen DP8390 eller kloner) i en Amiga ISA-port bryggkopplat via GG2-Bus+-kortet.

Den här drivrutinen ska stöda alla NE2000-kompatibla (inte NE1000-kompatibla) kort. Detta inkluderar kort som konfigureras med jumprar, kort som konfigureras via installationsdiskett (konfigurationen sparas i Flash-ROM) och jumperlösa ISA-PnP-kort utan bestämda inställningar.

MNI-drivrutinen hittar kortet automatiskt (IO-adress och IRQ). Manuell konfiguration stöds för tillfället inte. Om inga kort hittades inom det tillåtna IO- och IRQ-intervallet utför drivrutinen en PnP-sökning efter NE2000-kompatibla kort och konfigurerar och aktiverar alla funna kort under tidsperioden då drivrutinen är aktiv.

För närvarande kommer bara ett enda NE2000-kort i ISA-portarna att kännas igen.

Tillåtna MNI-argument:

- * Argumentet "FT" anger FIFO-tröskeln för lokal DMA. Möjliga värden är 0, 1, 2 eller 3. Förvalt är "FT=3". Du behöver vanligtvis inte ändra på det här värdet.
- * En av "WAIT=1", "WAIT=0" aktiverar/avaktiverar stödet för väntelägen på GG2-kort. Förvalt är "WAIT=1". "WAIT=0" använder kanske lite mindre CPU-tid med kort som är tillräckligt snabba.

1.113 Miami.guide/NODE_MNI_ZTAMSNNZ

z2-am7990.mni
=====

Drivrutin för kretsarna AM7990 (LANCE), AM79C90 (C-LANCE) och AM79C960 (PC-net) på Zorro-bussen. Kort som stöds för närvarande:

- * Ariadne (INTE Ariadne-II)
- * CEI/Ameristar A2065
- * Commodore A2065
- * ConneXion i A2065-kompatibilitetsläge

Viktig information för användare av ConneXionkort: se till att jumpern på kortet är inställd för A2065-kompatibilitet. Drivrutinen stöder för tillfället inte ConneXions ursprungsläge.

Viktig information för användare av A2065-kort: När du byter från SANA-II-drivrutin till MNI-drivrutin måste du först ändra inställningarna till MNI i Miami, spara inställningarna och sedan starta om datorn innan du kopplar upp dig igen. Detta är nödvändigt eftersom SANA-II-drivrutinerna för A2065 endast kan tas bort från minnet genom att starta om datorn.

Viktig information för användare av Ariadnekort: Din Amiga kanske redan har en Ariadne-drivrutin i "Sys:Expansion". Har den det måste du ta bort den drivrutinen och starta om din Amiga innan du kopplar upp dig med den nya MNI-drivrutinen. Annars kan din Amiga låsa sig.

Tillåtna MNI-argument:

- * Endast Ariadne: "MEDIA=AUTO" aktiverar automatisk mediadetektering (kabeltyp) och är det förvalda värdet. "MEDIA=10BASE2" väljer 10-Base-2, det vill säga, koaxialkabel, BNC eller "cheapernet". "MEDIA=10BASET" väljer 10-Base-T, det vill säga, kabel för RJ45 eller UTP.

1.114 Miami.guide/NODE_MNI_ZTDPETNZ

z2-dp8390.mni

=====

Drivrutin för kretsen DP8390 (och dess talrika kloner, till exempel integrerade kretsar som används i NE2000-kompatibla kort) på Zorrobussen. Korts som stöds för närvarande:

- * ASDG LanRover EB920
- * Hydra AmigaNet
- * Ariadne-II (INTE Ariadne)

Observera att under tiden som den här dokumentation skrevs var stödet för Ariadne-II avstängt i drivrutinen som följde med Miami 3.2, eftersom inget kort fanns tillgängligt för test. Stödet för Ariadne-II är implementerat och kommer att vara tillgängligt så fort det finns ett kort att testa.

Viktig information för användare av ASDG LanRover EB920: Kortet har en jumper som väljer interrupt (2 eller 6). Du måste använda ett MNI-argument som stämmer överens med dina jumperinställningar på kortet. Annars kommer din Amiga att krascha. Några EB920-kort har inget "Mac address ROM" på kortet. Om ditt kort tillhör dem kommer "Mac-adressen" som returneras av "Fråga drivrutinen" i "MNI-parametrar" att vara "00:00:00:00:00:00". I så fall måste du ange en godtycklig adress och kryssa för "ersätt". Se dokumentation för ditt EB920-kort för mer information om det här.

Viktig information för användare av Hydra AmigaNet: Med en relativt snabb CPU (68040 eller högre) får den här drivrutinen typiskt en väldigt hög överföringshastighet (> 800 kB/s) på lokala nätverk. Om du får låg överföringshastighet (500 kB/s eller mindre) på ett lokalt nätverk är det troligtvis fel på ditt Hydrakort. Tyvärr har väldigt många Hydrakort den här typen av problem. En låg hastighet beror alltså inte på ett fel i drivrutinen.

Tillåtna MNI-argument:

- * Argumentet "FT" anger FIFO-tröskeln för lokal DMA. Möjliga värden är 0, 1, 2 eller 3. Förvalt är "FT=2". Du behöver vanligtvis inte ändra på det här värdet.
- * Endast ASDG LanRover EB920: Antingen "INT=2" eller "INT=6". Det här argumentet måste stämma överens med jumperinställningarna på kortet.

1.115 Miami.guide/NODE_MNI_ZTMBESNFZ

z2-mb86950.mni

=====

Drivrutin för kretsen Fujitsu MB86950 Ethernet på Zorrobussen.

Kort som stöds för närvarande:

* QuickNet QN2000

Viktig information för användare av QuickNet QN2000: Kortet har en jumper på baksidan som ändrar produkt-ID mellan 1 och 2. Båda ID-numrena stöds, men ID 2 verkar fungera bättre. Det är därför rekommenderat att du om möjligt konfigurerar ditt kort för produkt-ID 2. För att göra detta, kontrollera först vilket produkt-ID du har (klicka på "Hitta kort" i Miami). Om produkt-ID är 1 ändra på jumpern, starta om din Amiga och försök igen. Alla kort kan inte ha produkt-ID 2. En del stöder endast produkt-ID 1 oberoende av inställningarna på jumpern.

Tillåtna MNI-argument: inga.

1.116 Miami.guide/NODE_MNI_ZTSMCNOCNZ

z2-smc91c90.mni
=====

Drivrutin för kretsen SMC91C90 (och dess efterföljare, till exempel SMSC LAN91C94) på Zorrobussen. Korts som stöds för närvarande:

* CEI/Ameristar A4066

Tillåtna MNI-argument: inga.

1.117 Miami.guide/NODE_DIALERLANG

Kommandospråk för uppringaren

Följande kommandon stöds för närvarande av uppringaren:

ABORT "text1", "text2", ...

Anger en lista med texter från modemmet som får Miami att helt avbryta uppringning, till exempel "NO DIALTONE".

ASKPASSWORD

Visar en dialogruta som frågar användaren efter lösenord.

DELAY sek

Väntar i angivet antal sekunder.

DIALNEXT "text1", "text2", ...

Anger en lista med texter från modemmet som får Miami att lägga på och ringa nästa nummer, till exempel "BUSY".

PARSEPASSWORD "sluttecken"

Läser in alla tecken från modemmet upp till, men inkluderar inte, <sluttecken> och ersätter det aktuella lösenordet med denna text. Detta kommando kan vara användbart för system med engångslösenord som skickar lösenordet för nästa session under inloggningen.

REDIAL "text1","text2",...

Anger en lista med texter från modemmet som får Miami att lägga på och ringa upp numret igen, till exempel "BUSY".

SAVECONFIG

Sparar de nuvarande inställningarna till disk. Detta kommando används vanligtvis efter PARSEPASSWORD för att spara inställningarna som innehåller det nya lösenordet.

SEND "text"

Skickar <text> till modemmet. Radmatning läggs inte till automatiskt. Miami känner igen följande vanliga kontrollsekvenser: \", \\", \r, \n. "\u" och "\p" stöds även och de skickar det aktuella användarnamnet respektive lösenordet.

SEENDBREAK

Skickar en avbrottssignal över serieporten. Denna används av några terminalservrar för att byta till kommandoläge.

SENDPAD "text",totallängd

Skickar <text> till modemmet, fyllt med mellanslag upp till längden <totallängd>. Exempel: 'SENDPAD "abc",5' skulle skicka "abc ".

SENDPASSWORD

Skickar det aktuella lösenordet, följt av "\r".

SENDUSERID

Skickar det aktuella användarnamnet följt av "\r".

TIMEOUT sek

Anger antal sekunder Miami ska vänta på text efter "WAIT" eller "WAITPPP" innan det ger upp.

WAIT "text"

Väntar på att "text" ska tas emot från modemmet.

WAITCONNECT

Väntar på att ett "CONNECT"-meddelande och den följande texten (vanligtvis uppkopplingshastighet) ska tas emot från modemmet. Detta är identiskt med 'WAIT "CONNECT"' förutom att Miami kopierar allt som följer "CONNECT"-meddelandet på samma rad till en intern buffert och visar senare det i status-arean. Med många modem låter detta dig se med vilken hastighet ditt modem kopplade upp.

WAITPPP

Vänta på att servern ska byta till PPP-läge.

Med kommandona "ABORT", "DIAL" och "DIALNEXT" kan du ange nyckelordet "TIMEOUT" (utan citationstecknena), istället för en text inom

citationstecken, till exempel

```
ABORT "NO CARRIER",TIMEOUT
```

Detta betyder att Miami kommer att avbryta uppringningsskriptet när det blir timeout. Andra val är att ringa upp igen eller ringa nästa nummer när det blir timeout.

1.118 Miami.guide/NODE_AREXX

ARExxgränssnitt

Namnet på Miamis ARExxport är "MIAMI.1". För tillfället stöder Miami alla standardkommandon för MUI-applikationer ("QUIT", "HIDE", "DEACTIVATE", "SHOW", "ACTIVATE", "INFO", "HELP") plus följande kommandon:

CHANGEDB

Säger till Miami att läsa in filen "ENVARC:MiamiChangeDB" för att uppdatera inställningarna. Se

Klientinställningar

för mer

information om hur du använder den här funktionen.

GETCONNECT

Returnerar uppkopplingssträngen som följde "CONNECT"-meddelandet från modemmet. Vanligtvis innehåller den här strängen en indikation på uppkopplingshastigheten.

GETCONNECTTIME

Returnerar antalet sekunder sedan Miami fick "CONNECT"-meddelandet från modemmet.

GETONLINETIME

Returnerar antal sekunder som Miami har varit uppkopplad i variabeln "RESULT".

GETSETTINGSNAME

Returnerar filnamnet på den nuvarande nuvarande inställningsfilen i variabeln "RESULT".

ISONLINE

Returnerar om Miami är uppkopplat eller inte i variabeln "RC". 1 returneras om Miami är uppkopplat och 0 om Miami inte är uppkopplat.

KILLGUI

Ikonifierar Miamis fönster och tar bort den aktuella GUI-motorn från minnet.

LOADSETTINGS file/a

Laddar den angivna inställningsfilen.

LOCKGUI

Låser användargränssnittet och visar en upptagen muspekare.
Anrop till den här funktionen nästlar.

OFFLINE

Lägger på och kopplar ned. Samma sak som att klicka på knappen
"Koppla ned".

ONLINE

Försöker koppla upp. Samma sak som att klicka på knappen
"Koppla upp".

QUITFORCE

Att använda kommandot "QUIT" från ett ARExxskript är det säkraste sättet att avsluta Miami eftersom Miami bara försöker koppla ned och avsluta om inga andra ARExxskript körs; detta för att undvika låsningar. Nackdelen med detta är att det kan bli problem om din ARExxkontroll är komplex eller innefattar flera ARExxskript (speciellt skript för tidigare händelser) och om ett eller flera skript fortfarande körs när "QUIT" kommandot anropas. Miami kommer då vägra att avsluta trots att det kanske är säkert att vänta och avsluta senare. Prova "QUITFORCE" i sådana fall; det tvingar Miami att vänta med att avsluta tills alla ARExxskript har avslutats. Varning: detta kommando kommer att låsa Miami om ett av ARExxskripten aldrig avslutar, till exempel på grund av en oändlig loop eller ett rekursivt anrop, så det är potentiellt farligt om ditt ARExxskript innehåller fel.

UNLOCKGUI

Låser upp användargränssnittet och tar bort den upptagna muspekaren och ersätter den med en normal om inte "LOCKGUI" har anropats flera gånger.

1.119 Miami.guide/NODE_ENVVARS

Miljövariabler

Tidszonsinformation

Tidszonsinformation

Användare behöver vanligtvis inte sätta några miljövariabler för att använda Miami. Icke desto mindre här är en lista med alla variabler som Miami använder om du skulle vilja göra manuella ändringar:

DOMAIN, DOMAINNAME

De här variablerna sätts automatiskt av Miami när Miami kopplar upp. De sätts till din aktuella domän (det vill säga, till den del av ditt värddamn som följer den första punkten ".").

HOME

Den här variabeln sätts automatiskt av Miami när Miami kopplar upp. Den sätts till hemkatalogen (som anges i Databas/användare) för användaren som står angivet på TCP/IP-sidan.

HOST, HOSTNAME

Dessa variabler sätts automatiskt av Miami när Miami kopplar upp. De sätts till ditt konfigurerade värddamn (för statiska värddamn) eller till värddamnet som stämmer överens med din IP-adress som hittades av Miami genom omvänd DNS-sökning. Om inget värddamn hittades blir dessa variabler satta till din IP-adress.

MagicWB

Om inget användargränssnitt är specificerat (av användaren i inställningsfilen eller i Env:MIAMI/GUI) kommer Miami att välja antingen "MUI" eller "MUIMWB" som förvalt användargränssnitt. "MUIMWB" används om variabeln "MagicWB" används, som indikerar att MagicWB har installerats.

REALNAME

Den här variabeln sätts automatiskt av Miami när Miami kopplar upp. Den sätts till det riktiga namnet som står angivet på TCP/IP-sidan.

SOCKETCONFIG

Den här variabeln sätts automatiskt av Miami när Miami kopplar upp. Den behövs för det fria emuleringsbiblioteket "socket.library" (för I-Net-225-kompatibel mjukvara) och den sätts så att biblioteket fungerar ordentligt.

TZ

Den här variabeln läser Miami för att hitta din tidszon. Den bör vara korrekt satt innan Miami installeras. Se

Tidszonsinformation
för mer information.

USERNAME

Den här variabeln sätts automatiskt av Miami när Miami kopplar upp. Den sätts till ditt användarnamn som står angivet på TCP/IP-sidan.

MIAMI/GUI

Den här variabeln ska innehålla namnet på den förvalda GUI-motorn (till exempel "MUI", "MUIMWB" eller "GTLayout"). Den sätts automatiskt vid installationen.

MIAMI/SSLIB

Den här variabeln behövs endast när du använder MiamiSSL och sätts automatiskt under installationen av MiamiSSL. Den ska innehålla namnet på SSL-krypteringsbiblioteket, det vill säga, antingen "Miami:Libs/miamisslntl.library" eller "Miami:Libs/miamisslusa.library".

1.120 Miami.guide/NODE_ENVVARS_TZ

Tidszonsinformation

Miljövariabeln TZ måste vara något av följande:

Under vintertid:

EST5

där EST är namnet på din tidszon och 5 är den negativa tidsskillnaden till UTC (det vill säga, om du är 2 timmar öster om UTC ska värdet vara -2, inte 2). I Europa, Asien och Australien är värdet vanligtvis 0 eller negativt, i Amerika är det positivt. Se nedan för exempel.

Under sommartid:

EST4EDT

EST, 4: betyder samma som ovan. EDT är namnet på din tidszon under sommartid. Se också till att ändra numret (4 i exemplet) med en timma, som krävs för sommartid.

Namnet på tidszonen spelar egentligen ingen roll i något av fallen. Det viktiga är att numret är rätt och att numret börjat exakt på den fjärde bokstavspositionen.

Det är inte korrekt att sätta en så kallad RFC-tidssträng i Env:TZ, det vill säga, "EST", "EST (-0500)" eller "-0500" kommer inte att fungera.

Några exempel:

	vinter	sommar
USA:s västkust	PST8	PST7PDT
USA:s östkust	EST5	EST4EDT
Storbritannien	WET0	WET-1WEDT
större delen av		
Västeuropa (Sverige)	MET-1	MET-2MEDT

1.121 Miami.guide/NODE_EXCONFIG

Byta inställningar

Miamiinställningarna sparas i en IFF-fil i ett format som för närvarande är medvetet odokumenterat. Miami låter dig dock importera och exportera inställningar på en mängd olika sätt:

Distributionsformat	Importerera/exportera inställningar
Byta lösenord	Byta lösenordsfiler
Klientinställningar	Egna inställningar för klienter

1.122 Miami.guide/NODE_EXCONFIG_DIST

Distributionsformat
=====

Miami låter dig exportera inställningar till ett ASCII-format (textformat) som lämpar sig för distribution, till exempel för att ladda upp det till Aminet, eller för att ge till någon annan användare som har konto hos samma Internetleverantör. Det kan även användas av Internetleverantörer för att konfigurera Miamis inställningar i förväg åt den nya användaren.

ASCII-filens format innehåller ett huvud, följt av ett varierande antal parametrar.

När du exporterar filer tar Miami endast med de parametrar som är relaterade till nätverksleverantören, men inte de som är relaterade till den individuella användarens systeminställningar eller som är säkerhetsrelaterade på något sätt. Det betyder att du på ett säkert sätt kan exportera dina inställningar och ge filen till en annan användare utan att behöva lämna ut känslig information som lösenord.

När Miami importerar filer stöds användarrelaterad information som lösenord för att nätverksleverantörer ska kunna skriva installationsskript som frågar efter användarnamn och lösenord och som skapar en inställningsfil för Miami som innehåller all information som krävs av Miami.

För att få en idé om hur ASCII-filen ser ut, exportera dina nuvarande inställningar till ASCII. Det generella formatet är:

- * ett huvud på två rader. Varje rad börjar med tecknet "\$".
Ändra inte det här huvudet.
- * ett varierande antal rader som börjar med ";". Dessa rader är kommentarer och kan ändras fritt.
- * ett varierande antal rader som anger parametrar.

De flesta parametrar anges på en enkel rad. Dessa rader ser ut så här:

PARAMETER=värde

En del parametrar (till exempel uppringningsskriptet) kräver flera rader. I det här fallet ser det ut som följer:

```
PARAMETER=%  
första värdet  
andra värdet  
tredje värdet  
%
```

Det betyder att tecknet "%" indikerar en flerradsparameter och ett till "%" ensamt på en rad indikerar slutet på listan med värden.

Ordningen på parametrarna är valfri. Du bör inte anta att Miami lagrar parametrarna i en speciell ordning.

Lista med tillåtna parametrar: Ett (m) indikerar en flerraders parameter. Ett (i) indikerar att parametern endast importeras men aldrig exporteras. Ett (r) indikerar att den här funktionen endast finns tillgänglig i den registrerade versionen och ignoreras i den oregistrerade versionen. "(m)", "(i)" och "(r)" finns inte i ASCII-filen.

Värden som anges som "A / B" betyder att värdet är en enkel bokstav, antingen "A" eller "B".

DEVNAME= (i)
drivrutinsnamn

UNIT= (i)
enhetsnummer

BAUD= (i)
serieporthastighet

PROTOCOL=
P / S (ppp eller slip)

FLOWCONTROL= (i)
H / S (hårdvaru- (RTS/CTS) eller mjukvaru-handskakning
(Xon/Xoff))

EOFMODE= (i)
Y / N / A (ja / nej / auto)

SERMODE=
8N1 / 7E1 / 7O1

MTU=
heltal

IPTYPE=
D / S (dynamisk eller statisk)

IP=
1.2.3.4

CD= (i)
Y / N (använd CD: ja / nej)

BOOTP=
Y / N (använd BootP: ja / nej)

INACTIVITY=
N / I / P (inaktivitetstyp: ingen, ICMP, PPP)

INACTIVITYDELAY=
minuter

PAPNAME= (i)
användarnamn

PAPPWD= (i)
lösenord

PAPSAME=
Y / N

CALLBACKTYPE= (r)
NONE / CBCPFIXED / CBCPVARIABLE

CALLBACKPHONE= (i) (r)
telefonnummer

CALLBACKMINDELAY= (r)
heltal

CALLBACKMAXDELAY= (r)
heltal

ACCM=
000a0000

VJC=
Y / N

QUICKRECONNECT=
Y / N

TERMREQ=
Y / N

DIALNAME= (i)
användarnamn

DIALPWD= (i)
lösenord

INITSTRING= (i)
modeminitieringskommando

DIALPREFIX= (i)
uppringningsprefix

DIALSUFFIX= (i)
uppringningssuffix

DIALSCRIPT= (m)
uppringningsskript

DIALNUMBERS= (i) (m)
telefonnummer

DIALMAXREPEAT=
max_upprepning

DIALREPEATDELAY=
repetitionskaus

DIALREDIALDELAY=
uppringningskaus

HOSTDYNAMIC=
Y / N (dynamiskt värddnamn: ja / nej)

HOSTNAME= (i)
värddnamn

REALNAME= (i)
riktigt_namn

USERNAME= (i)
användarnamn

DOICMP=
Y / N

FAKEIP=
Y / N

TTCP= (r)
Y / N

DBHOSTS= (m)
värddatabas

DBNETWORKS= (m)
nätverksdatabas

DBDOMAINS= (m)
domändatabas

DBDNSSERVERS= (m)
DNS-serverdatabas

1.123 Miami.guide/NODE_EXCONFIG_PASSWORDS

Byta lösenord

=====

Miami låter dig fritt importera och exportera alla filer från db-lådorna från Unix/AmiTCP med ett undantag: filen passwd kan importeras men lösenorden rensas i processen och måste manuellt återanges i Miami.

Anledningen för detta är att AmiTCP (åtminstone upp till version 4.3) använder DES-algoritmen för lösenordskryptering. DES är kryptografiskt sett en stark krypteringsalgoritm som lyder under amerikanska exportlagar. Ett program som implementerar DES får ej exporteras utanför USA utan individuellt tillstånd och den amerikanska regeringen utfärdar för tillfället inte sådana tillstånd.

Det resulterar i att all slags export av AmiTCP från USA är olaglig. Detta inkluderar även att ladda ned AmiTCP-arkivet från en ftp-server i USA till en dator utanför USA. På grund av detta får inte AmiTCP laddas upp till all Aminet-speglar, vilket försvårar tillgängligheten för AmiTCP.

För Miami skulle det ha varit ännu värre: eftersom jag utvecklar Miami i USA (inte i Finland som NSDi) skulle jag inte fått tillstånd att skicka Miami till någon utanför USA, oberoende av distributionssätt. Jag har därför beslutat att inte använda DES i Miami utan istället använda en annan krypteringsalgoritm som inte lyder under amerikanska exportlagar.

Miami använder sig av en itererad version av MD5 för lösenordskryptering. Denna algoritm är kryptografisk stark, det vill säga, ej forcerbar förutom vid grundlig sökning precis som DES. Eftersom MD5 är, olikt DES, en envägsalgoritm kan den inte dekrypteras och lyder därför inte under amerikanska exportlagar.

Det betyder att det är helt lagligt att importera och exportera Miami till och från USA, ladda upp Miami till Aminetspeglar och andra ftp-platser och att använda Miami i USA och andra länder (förutom om något land förbjuder användning av MD5).

Jag är ledsen för problemen som det här orsakar för användare som hanterar flera och/eller stora lösenordsfiler, men jag kan inte komma på något annat sätt att lösa situationen.

1.124 Miami.guide/NODE_EXCONFIG_CLIENTS

Klientinställningar

=====

En del TCP/IP-klienter som AmiTalk kräver ändringar i inställningsdatabasen som många protokollstackar lagrar i lådan db. Vanligtvis måste poster läggas till filerna services eller inetd.conf.

Med Miami kan du göra nödvändiga ändringar direkt genom det grafiska användargränssnittet, det vill säga, välja sidan "Databas", den korrekta avdelningen (till exempel "tjänster") och lägga till de poster du behöver.

I några situationer kan det vara mer bekvämt att automatisera den här processen, till exempel låta installationsskriptet för TCP/IP-klienten göra nödvändiga ändringar av sig självt, utan att behöva störa användaren. Med Miami fungerar detta på följande sätt:

- * Först måste du lägga till en rad till filen EnvArc:MiamiChangeDB som ser ut som följer:

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

 eller

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

 När Miami startas läser det innehållet av den här filen in (om den existerar), uppdaterar inställningarna och sparar de resulterande inställningarna.
- * Om Miami körs när klienten installeras och du vill att Miami ska uppdatera sina inställningar direkt ska du skicka ARexxkommandot "CHANGEDB" till Miami efter att ha ändrat ovanstående fil.

Du kan lägga till poster till alla Miamis databastabeller på det här sättet. För säkerhets skull ändras bara de tabeller som vanligtvis används av klienter ("inetd" och "tjänster") direkt av Miami. Om applikationer försöker ändra några andra tabeller (till exempel den känsliga tabellen "användare") kommer Miami att visa ett dialogfönster och vänta på en bekräftelse från användaren, efter att ha fått kommandot "CHANGEDB".

Sammanfattningsvis: i ditt installationsskript ska du ha följande rader för att automatiskt konfigurera Miami för din klient:

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

Om Miami körs uppdateras ändringarna automatiskt. Annars kommer Miami att hämta ändringarna nästa gång det startas.

1.125 Miami.guide/NODE_UTILITY

Nyttoprogram

MiamiArp

MiamiArp

MiamiFinger

MiamiFinger

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

1.126 Miami.guide/NODE_UTILITY_ARP

MiamiArp

=====

Visar och kontrollerar adressomvandling.

Användning:

MiamiArp värddamn

Visar aktuell ARP-post för <värddamn>

MiamiArp [-n] -a

Visar alla aktuella ARP-poster. Om "-n" anges kommer alla poster

att visas numeriskt istället för symboliskt.

MiamiArp -d värddamn

Tar bort ARP-post för <värddamn>

MiamiArp -s värddamn hv_adr [temp] [pub]

Skapar en ARP-post för <värddamn> med hårdvaruadressen <hv_adr>. Posten är permanent såvida inte argumentet "temp" anges. Om argumentet "pub" anges kommer systemet att agera som en ARP-server för den angivna värden.

MiamiArp -f filnamn

Läser och utför kommandon från filen <filnamn>.

1.127 Miami.guide/NODE_UTILITY_FINGER

MiamiFinger

=====

Visar information om systemanvändare.

Användning: MiamiFinger [-l] [användare][@datornamn]

Argument:

-l

Visar långt format (till andra datorers fingerdemon skickas modifieraren "/w").

Om inget datornamn anges antas "localhost".

Om en användare anges visas information om denna användare. Annars visas förvald information för den fingerdemon som uppkopplingen sker mot. I många fall är detta generell systeminformation eller de användare som för tillfället är inloggade.

Den här implementationen av MiamiFinger stöder T/TCP för snabbare sökningar.

1.128 Miami.guide/NODE_UTILITY_IFCONFIG

MiamiIfConfig

=====

Konfigurerar nätverksgränssnitt.

OBS: de flesta argumenten till MiamiIfConfig bör inte användas med Miami för tillfället, eftersom Miami normalt sätter alla värden korrekt. Lek inte med det här programmet; du bör veta precis vad du gör innan du försöker ändra några av gränssnittsparametrarna.

De enda argument som egentligen är användbara är "up" och "down", för att temporärt markera att ett gränssnitt inte är tillgängligt. Observera att detta inte gör att modemmet lägger på luren. I övrigt bör du förmodligen bara använda MiamiIfConfig för att undersöka gränssnittsparametrarna, inte för att ändra dem.

```
Användning: MiamiIfConfig gränssnitt [alias | -alias] [af [adress
[dest_adr]] [up] [down] [netmask mask]] [metric n] [arp | -arp]
[broadcast adress] [link0 | -link0] [link1 | -link1] [link2 | -link2]
```

Argument:

gränssnitt

För tillfället antingen "lo0" eller "mi0"

alias/-alias

Betrakta den angivna adressen som ett alias för den existerade adressen; det vill säga, ersätt inte en befintlig adress.

af

Adressfamilj: endast "inet" stöds för tillfället.

adress

En adress på protokollnivå. För adressfamiljen "inet" är detta en IP-adress i punktnotation (till exempel 123.45.67.89).

dest_adr

Destinationsadress på protokollnivå. Detta används bara för "punkt till punkt"-enheter.

up/down

Markera att gränssnittet är tillgängligt ("up") eller inte tillgängligt ("down").

netmask

Ändrar nätmask för det här gränssnittet.

metric

Ändrar prioritet för det här gränssnittet. Detta har ingen effekt för en stack som Miami som bara stöder ett gränssnitt.

arp/-arp

Sätter på eller stänger av ARP för det här gränssnittet. Det här argumentet bör inte användas med Miami. Använd Miamis GUI för att ange typ av adressomvandling.

broadcast

Sätter broadcastadress för det här gränssnittet.

linkx/-linkx

Sätter eller återställer länknivåflaggorna 0, 1 eller 2. De här flaggorna används för tillfället inte av Miami.

1.129 Miami.guide/NODE_UTILITY_MAPMBONE

MiamiMapMBone
=====

Kartlägger multicastkopplingar.

Användning: MiamiMapMBone [-d felsökningsnivå] [-f] [-g] [-r antal_försök] [-t väntetid] [startrouter]

MiamiMapMBone försöker visa alla multicastroutrar som kan nå från angiven startrouter. Om startrouter inte anges används "localhost".

Argument:

- d felsökningsnivå
Sätter felsökningsnivå. När felsökningsnivån är större än det förvalda värdet 0 visas fler felsökningsmeddelanden.
- f
Aktiverar rekursiv sökning av närliggande multicastroutrar. När startrouter inte anges är det förvalt att använda rekursiv sökning.
- g
Ritar grafer i GraphEd-format.
- n
Gör inga DNS-sökningar för namn på multicastroutrar.
- r antal_försök
Sätter antal försök för att nå närliggande multicastroutrar. Förvalt är 1 försök.
- t väntetid
Sätt antal sekunder att vänta på svar från grannroutrar innan ett nytt försök görs. Förvald väntetid är 2 sekunder.

1.130 Miami.guide/NODE_UTILITY_MRINFO

MiamiMRInfo
=====

Visar konfigurationsinformation för en multicastrouter.

Användning: MiamiMRInfo [-d felsökningsnivå] [-r antal_försök] [-t timeout] [multicastrouter]

MiamiMRInfo försöker visa konfigurationsinformationen för den angivna multicastroutern. Om ingen router anges används "localhost".

Argument:

- d felsökningsnivå
Sätter felsökningsnivå. När felsökningsnivån är större än det förvalda värdet 0 visas fler felsökningsmeddelanden.
- r antal_försök
Sätter antal försök att nå närliggande multicastrotrar.
Förvalt är 3 försök.
- t timeout
Sätter antal sekunder att vänta på svar från grannroutrar innan ett nytt försök görs. Förvald väntetid är 4 sekunder.

1.131 Miami.guide/NODE_UTILITY_MROUTED

MiamiMRouteD
=====

Demon för multicastroutning.

Användning: MiamiMRouteD [-p] [-c konfigurationsfil] [-d felsökningsnivå]

MiamiMRouteD är ett program du kan behöva ha körande i bakgrunden ("run MiamiMRouteD") för att ta emot och vidarebefordra multicastflöden. Se nedan för en mer detaljerad förklaring.

Argument:

- p
Startar MiamiMRouteD i ej rensande läge. Det här argumentet bör bara användas vid testning.
- c konfigurationsfil
Anger konfigurationsfil att använda. Den förvalda konfigurationsfilen är Miami;MiamiMRouteD.config.
- d felsökningsnivå
Sätter felsökningsnivå. När felsökningsnivån är större än det förvalda värdet 0 visas fler felsökningsmeddelanden.

MiamiMRouteD är ett mycket komplext och kraftfullt program som låter dig ta emot och vidarebefordra multicastflöden. Programmet konfigureras via en separat konfigurationsfil vars format delvis beskrivs nedan. Eftersom Miami bara stöder ett gränssnitt är det bara några funktioner i MiamiMRouteD som är meningsfulla att använda med Miami; förmodligen är det få användare som över huvud taget behöver det.

De två vanligaste konfigurationerna är:

- * Multicastflödet kommer via ett broadcast- eller multicastkapabelt gränssnitt, till exempel Ethernet eller Arcnet. Kör inte MiamiMRouteD i detta fall. Aktivera istället multicast i Miami på

sidan "Gränssnitt".

- * Multicastflöde kommer genom en IP-tunnel, till exempel via PPP från din Internetleverantör. I det här fallet ska du stänga av multicast i Miami för ditt PPP/(C)SLIP-gränssnitt, konfigurera MiamiMRouteD för en tunnel till din leverantör (se nedan) och köra MiamiMRouteD efter Miami har startats.

Konfigurationsfilen för MiamiMRouteD är en vanlig textfil. Varje rad kan innehålla ett kommando. Det enda intressanta kommandot är för tillfället "tunnel", som låter dig konfigurera en IP-tunnel för att skicka och ta emot multicastflöden. Den viktiga delen av syntaxen är:

```
tunnel <lokaladress> <fjärradress>
```

För <lokaladress> kan du ange en IP-adress eller ett gränssnittsnamn (för Miami är detta alltid "mi0"). <fjärradress> är värden på den andra sidan av multicasttunneln, till exempel:

```
tunnel mi0 1.2.3.4
```

etablerar en multicasttunnel från värden 1.2.3.4.

1.132 Miami.guide/NODE_UTILITY_MTRACE

MiamiMTrace

=====

Visar multicastväg från en källa till en mottagare

Användning: MiamiMTrace [-g brygga] [-i gr_adr] [-l] [-M] [-m max_hopp] [-n] [-p] [-q antal_frågor] [-r svar_dest] [-s] [-S stat_int] [-t ttl] [-v] [-w väntetid] källa [mottagare] [grupp]

MiamiMTrace är ett verktyg som är mycket likt MiamiTraceRoute, men arbetar med multicastadresser istället för unicastadresser. Se

MiamiTraceRoute

för mer information om TraceRoute. <grupp> anger multicastadress att använda. <källa> och <mottagare> är unicastadresser som anger start- och slutpunkt för multicastvägen som ska spåras. Om <grupp> inte anges används 224.2.0.1. Om <mottagare> inte anges används "localhost".

Argument:

-g brygga

Skickar spårningsfrågan via unicast direkt till angiven multicastrouter istället för att skicka frågan via multicast. Denna router måste vara den sista hopproutern på vägen från den avsedda källan till mottagaren.

-i gr_adr

Använder den angivna adressen som den lokala gränssnittsadressen (på en värd med flera hem; som är "multi homed") för att skicka spårningsfrågan från och som förvalt värde för mottagare och svarsdestination.

-l

Visar pakethastighet och förluststatistik för multicastvägen var tionde sekund (se även "-S stat_int").

-M

Skickar alltid svaret via multicast istället för att försöka med unicast först.

-m max_hopp

Sätter maximalt antal hopp som ska spåras från mottagaren tillbaka till källan. Förvalt är 32 hopp (oändligt för routningsprotokollet DVMRP).

-n

Skriver ut hoppadresser numeriskt istället för symboliskt och numeriskt (detta undviker en adress-sökning på en namnserver för varje router som hittas på vägen).

-q antal_frågor

Sätter maximalt antal frågeförsök för ett hopp. Förvalt är 3.

-p

Lyssnar passivt på multicastsvar från spårningar som initieras av andra. Detta fungerar bäst när det körs på en multicastrouter.

-r svar_dest

Skickar spårningssvaret till angiven värd istället för den värd på vilken MiamiMTrace körs eller till en multicastadress annan än den som är registrerad för det här syftet (224.0.1.32).

-s

Skriver utmatningen i ett kort format som bara inkluderar multicastsökvägen och inte pakethastighet eller förluststatistik.

-S stat_int

Sätter intervall mellan spårningar för att samla statistik. Förvalt är 10 sekunder.

-t ttl

Sätter ttl (time-to-live, eller antal hopp) för multicastspårningar och svar. Förvalt är 64, utom för lokala frågor till multicastgruppen "alla routrar", där förvalt värde är 1.

-v

Utförligt läge; visar hopptider för den initiala spårningen och statistikvisningen.

-w väntetid
Sätter tid att vänta på ett spårningssvar till angivet antal sekunder.

1.133 Miami.guide/NODE_UTILITY_NETSTAT

MiamiNetStat
=====

Visar konfigurationsparametrar och statistik. Programmet är nästan funktionellt identiskt med den version av "netstat" som följer med 4.4BSD, men har några ytterligare funktioner för att visa länknivåstatistik.

- * MiamiNetStat [-AaDnN] [-f adressfamilj]
- * MiamiNetStat [-dimnNrs] [-f adressfamilj]
- * MiamiNetStat [-dnN] [-] [-I gränssnitt]
- * MiamiNetStat [-s] [-] [-L gränssnitt]
- * MiamiNetStat [-s] [-g]
- * MiamiNetStat [-p protokoll]

MiamiNetStat visar det symboliska innehållet av diverse nätverksrelaterade datastrukturer. Det finns ett antal utmatningsformat för den information som presenteras, beroende på vilka argument som anges.

Den första kommandoformen visar en lista på de aktiva anslutningarna för varje protokoll.

Den andra formen visar innehållet i en av de andra nätverksdatastrukturerna som anges som argument.

Den tredje formen visar information om pakettrafiken på det angivna nätverksgränssnittet.

Den fjärde formen visar information om länknivåkonfiguration eller, om argumentet "-s" anges, länknivåstatistik för det angivna nätverksgränssnittet.

Den femte formen visar information om virtuella gränssnitt (för multicasting) och statistik om multicastingstrutning.

Den sjätte formen visar statistik om det angivna protokollet.

Argument:

-A
Med det förvalda utmatningsformatet: visar adresserna för alla

protokollkontrollblock för anslutningar; används för felsökning.

-a

Med det förvalda utmatningsformatet: visar läget för alla kontakter; normalt visas inte kontakter som används av serverprocesser.

-d

När ett gränssnitt visas (argument "-i" eller "-I"): visar antalet tappade paket.

-D

Med det förvalda utmatningsformatet: visar det totala antalet överförda byte för varje aktiv TCP-uppkoppling.

-f adressfamilj

Begränsar statistik eller rapporter för adresskontrollblock till de för den angivna adressfamiljen. Adressfamiljen "inet" är den enda som känns igen för tillfället.

-g

Visar den virtuella gränssnittstabellen och multicastroutningstabellen. Tillsammans med argumentet "-s" visar det här argumentet statistik för multicastroutning. Båda dessa argument är bara meningsfulla att använda om MiamiMRouteD kör.

-I gränssnitt

Visar information om angivet gränssnitt.

-i

Visar läget på de gränssnitt som har konfigurerats.

-m

Visar statistik över minneshanteringen (nätverket hanterar en privat uppsättning av minnesbuffrar).

-n

Visar nätverksadresser numeriskt (normalt tolkar MiamiNetStat adresser och försöker visa dem symboliskt). Detta argument kan användas med alla utmatningsformat.

-N

Visar bara nätverksadresser symboliskt om det symboliska namnet är tillgängligt utan att göra en DNS-sökning. Annars visas nätverksadressen numeriskt. Detta argument kan användas med alla utmatningsformat.

-p protokoll

Visar statistik om angivet protokoll, vilket antingen är ett välkänt namn på ett protokoll eller ett alias för det. Ett tomt svar betyder normalt att det inte finns något intressant att rapportera. Programmet påpekar om angivet protokoll är okänt eller om det inte finns någon statistikfunktion för det.

-r

Visar routningstabellerna. Om argumentet "-s" också anges visas

routningsstatistik istället.

-s

Visar statistik för varje protokoll. Om det här argumentet anges två gånger visas bara värden som är skilda från noll.

Det förvalda utmatningsformatet, för aktiva kontakter, visar lokal- och fjärradresser, storlek på in- och utbuffrar (i byte), protokoll, och intern status för protokollet. Adresserna visas på formatet "värd.port" eller "nätverk.port" om en kontakts adress anger ett nätverk men ingen specifik värdadress. När de är kända visas värd- och nätverksadresserna symboliskt enligt databaserna "värdar" och "nätverk". Om ett symboliskt namn för en adress inte är känd, eller om argumentet "-s" har angetts, visas adressen numeriskt enligt adressfamiljen.

Utmatningsformatet för gränssnitt visar en tabell med kumulativ statistik över antalet överförda paket, antal fel och kollisioner. Nätverksadresserna för gränssnittet och maximal paketstorlek ("MTU") visas också.

Utmatningsformatet för routningstabeller visar de tillgängliga vägarna och deras status. Varje väg består av en destinationsvärd eller -nätverk samt en brygga att använda när paket vidarebefordras. Flaggfältet visar en uppsättning information om vägen i form av binära flaggor.

1

RTF_PROTO1 Protokollspecifik routningsflagga ett (används för tillfället inte).

2

RTF_PROTO2 Protokollspecifik routningsflagga två (används för tillfället inte).

3

RTF_PROTO3 Protokollspecifik routningsflagga tre (för TCP: vägens väntetid löper ut).

C

RTF_CLONING Genererar nya vägar vid användning.

D

RTF_DYNAMIC Skapad dynamiskt (genom omdirigering).

G

RTF_GATEWAY Destinationen kräver vidarebefordring av en mellanhand.

H

RTF_HOST Värdpost (nätpost annars).

L

RTF_LLINFO Giltig översättning från protokolladress till länkadress.

M

RTF_MODIFIED Modifierad dynamiskt (genom omdirigering).

P

RTF_PRCLONING Klonar vägar för användning av protokoll.

R

RTF_REJECT Vård eller nät kan inte nås.

S

RTF_STATIC Har lagts till manuellt.

U

RTF_UP Vägen kan användas.

W

RTF_WASCLONED Vägen skapades genom kloning av en annan väg.

X

RTF_XRESOLVE Extern demon översätter protokolladress till länkadress.

Direkta vägar skapas för varje gränssnitt som är kopplat till den lokala värden; fältet "gateway" för sådana poster visar adressen för det utåtgående gränssnittet. Fältet "refcnt" visar antalet aktiva användningar av vägen. Uppkopplingsorienterade protokoll håller sig normalt till en enstaka väg för hela uppkopplingen medan ej uppkopplingsorienterade protokoll erhåller en väg i samband med att de skickar till samma destination. Fältet "use" visar antalet paket som har skickats via den aktuella vägen. Fältet "interface" anger vilket nätverksgränssnitt som användes för vägen.

Med argumentet "-L" visar MiamiNetStat information om länknivåkonfigurationen, såsom den aktuella statusen på PPPs underprotokoll IDCP och LCP för det angivna gränssnittet.

Med argumentkombinationen "-sL" visar MiamiNetStat länknivåstatistik, inklusive information om olika typer av paket samt kontrollsummefel för angivet gränssnitt.

För tillfället stöder Miami bara två gränssnitt:

lo0

Det lokala loopgränssnittet.

mi0

Gränssnittet för PPP/(C)SLIP som använder Miamis inbyggda gränssnittsdrivrutin eller det aktuella SANA-II-gränssnittet.

1.134 Miami.guide/NODE_UTILITY_PING

MiamiPing

=====

Skickar paket till nätverkssvärdar och lyssnar på deras svar.

Användning: MiamiPing [-Rdfnqrv] [-c antal] [-i paus] [-l förladdning] [-p mönster] [-s paketstorlek] värddamn

Argument:

-c antal

Sluta efter att ha skickat och tagit emot <antal> paket.

-d

Sätter argumentet SO_DEBUG för den anslutning som används.

-f

Skickar paket så fort som de kommer tillbaka eller minst hundra gånger per sekund. Varje skickat paket visas med en punkt ("."), medan varje svar visas med ett baksteg. Detta visar snabbt hur många paket som tappas. OBS: att använda det här argumentet för att överbelasta värdar är olagligt.

-i paus

Vänta <paus> sekunder mellan varje paket som skickas. Förvalt är att vänta en sekund mellan varje paket. Det här argumentet kan inte användas tillsammans med "-f".

-l förladdning

Skicka <förladdning> paket så fort som möjligt. Skicka därefter paket i normal hastighet.

-n

Visa bara numerisk utmatning.

-p mönster

Du kan ange upp till sexton extrabyte för att fylla upp de skickade paketen. Detta är användbart för att undersöka databeroende problem i ett nätverk. Till exempel kommer "-p ff" att fylla det skickade paketet med binära ettor.

-q

Minimal utmatning. Inget visas utom summeringstexterna vid start och avslut.

-R

Spela in vägen. Inkludera flaggan RECORD_ROUTE i skickade paket och visa vägbuffert i mottagna paket. Observera att IP-huvudet bara har plats för nio sådana här vägar. Många värdar ignorerar eller tar bort den här flaggan.

-r

Gå förbi de normala routningstabellerna och skicka paket direkt till en värd på ett "anslutet" nätverk. Om värden inte finns på ett direkt anslutet nätverk returneras ett fel. Det här argumentet kan användas för att pinga en lokal värd genom ett gränssnitt som inte har någon väg igenom sig (till exempel efter det att gränssnittet togs bort av routed).

-s paketstorlek

Ange antalet databyte som ska skickas. Förvalt är 56, vilket

motsvarar 64 ICMP-databyte när det kombineras med ICMP-huvudet på 8 byte.

-v

Utförlig utmatning. Andra ICMP-paket än pingsvar som tas emot visas.

1.135 Miami.guide/NODE_UTILITY_REMIND

MiamiRemind
=====

En del användare tycker att den automatiska varningen och nedkopplingen efter trettio respektive sextio minuter i demoversionen av Miami är en användbar funktion för att spara på telefon- eller ISP-kostnader.

MiamiRemind är ett verktyg som erbjuder den här typen av funktionalitet i den registrerade versionen av Miami. Utöver den enkla varningen eller nedkopplingen erbjuder det flera andra användbara funktioner:

- * Antalet varningar, så väl som intervallet mellan kommande varningar, kan konfigureras fritt.
- * Det är möjligt att koppla ned efter en viss tid, att bara visa en begränsat antal varningar (utan att koppla ned) eller att kontinuerligt visa regelbundna varningar.
- * Utöver att ange fasta tidsintervall är det möjligt att visa varningar efter en viss tid av inaktivitet på länken. Båda typerna av varningar (varningar efter fasta intervall och varningar efter inaktivitet) kan användas samtidigt.
- * Genom att använda en inaktivitetsperiod tillsammans med nedkopplingsfunktionen är det möjligt att "koppla ned vid inaktivitet", vilket många användare har efterfrågat.

Termen "inaktivitet" är svår att definiera för en TCP/IP-uppkoppling. Den förvalda definitionen i MiamiRemind är "brist på TCP-trafik". Med denna definition använder MiamiRemind extremt lite resurser, både när det gäller CPU-tid och minne.

För användare som behöver en mer sofistikerad definition av "inaktivitet" erbjuder MiamiRemind en tolk och kompilator för uttryck, identisk med den som finns i MiamiTCPDump. Du kan alltså använda uttryck som

```
"(tcp[13] & 3 != 0) or udp"
```

Ovanstående uttryck skulle betrakta alla TCP-"SYN"-paket, alla TCP-"FIN"-paket och alla UDP-paket som "aktivitet". Alla andra paket ignoreras.

Uttryckstolken och -kompilatorn kräver `miamibpf.library` och `miamipcap.library`, och kräver alltså lite mer minne och CPU-tid än den inbyggda definitionen av TCP-trafik.

```
Användning: MiamiRemind [-f fast_tid_spec]
                  [-i inaktivitetstid__spec]
                  [-p pcap_spec]
```

Argumentet `"-f"` anger fasta intervall, det vill säga de som börjar räknas när `MiamiRemind` startas och som inte tar hänsyn till aktiviteten på länken. Det förvalda är att inte använda några fasta intervall.

Argumentet `"-i"` anger inaktivitetsintervall. Dessa nollställs varje gång ett paket som räknas som aktivitet skickas eller tas emot via länken. Det förvalda är att inte använda några inaktivitetsintervall.

Argumentet `"-p"` anger inaktivitetsuttryck i `MiamiPCap`format (se exemplet ovan). Uttrycket bör anges inom citationstecken (`"`). Om det här argumentet anges använder `MiamiRemind` `miamipcap.library` och `miamibpf.library` för att tolka, kompilera och utvärdera uttrycket. Annars används den inbyggda definitionen på TCP-trafik och de två biblioteken används inte.

`"tid_spec"` (för argumenten `"-f"` och `"-i"`) är en sträng som består av tal som representerar tidsintervall (i antal minuter), separerade av kommatecken (`"`,`"`).

Varje tidsintervall i strängen representerar pausen mellan två efterföljande händelser.

En "händelse" är oftast ett varningsfönster. Det är dock även möjligt att ange ett bokstavsprefix före talet. Prefixet `"D"` anger att `MiamiRemind` ska koppla ned länken. Prefixet `"L"` anger att det efterföljande intervallet ska upprepas - så länge som behövs - för att definiera en sekvens av händelser.

Exempel:

```
MiamiRemind -f 30,D30
```

Detta är identiskt med hur demoversionen av `Miami` uppträder. Först visas en varning efter trettio minuter och nedkoppling sker efter ytterligare trettio minuter.

```
MiamiRemind -f 30,20,L10
```

Visa en varning efter trettio minuter, sedan efter tjugo minuter och efter det var tionde minut. Koppla inte ned länken.

```
MiamiRemind -f 60,60 -i L10
```

Visa en varning efter sextio minuter och sedan efter ytterligare sextio minuter. Därefter visas bara varningar efter var tionde minut av inaktivitet (ingen TCP-trafik) på länken.

```
MiamiRemind -i D30
```

Koppla ned länken efter trettio minuter av inaktivitet (ingen TCP-trafik).

```
MiamiRemind -i D20 -p "tcp or udp"
```

Koppla ned länken efter tjugo minuters inaktivitet. "Inaktivitet" gäller för TCP- eller UDP-trafik.

MiamiRemind avslutas automatiskt när gränssnittet kopplas ned (oberoende av orsak), när Miami försöker avsluta, när programmet tar emot en Ctrl-C-signal eller när inga intervall är aktiva.

Det enklaste sättet att använda MiamiRemind är att starta det direkt från Miami när Miami kopplar upp, till exempel som "run >nil: Miami: MiamiRemind [argument]" i ett skalskript som startas från Miami (konfigureras i "Händelser" / "Uppkoppling").

1.136 Miami.guide/NODE_UTILITY_RESOLVE

```
MiamiResolve  
=====
```

Omvandlar ett värddamn till en IP-adress eller en IP-adress till ett värddamn.

Användning:

```
MiamiResolve ip_adress
```

Omvandlar en IP-adress och visar tillhörande värddamn med alla tillhörande IP-adresser.

```
MiamiResolve värddamn
```

Omvandlar ett värddamn och visar tillhörande värddamn med alla tillhörande IP-adresser.

```
MiamiResolve -s portnummer
```

Omvandlar ett portnummer och visar alla tillhörande tjänster och portnummer.

```
MiamiResolve -s tjänst
```

Omvandlar en tjänst och visar alla tillhörande tjänster och portnummer.

1.137 Miami.guide/NODE_UTILITY_ROUTE

```
MiamiRoute  
=====
```

Manipulerar routningstabellerna manuellt.

Användning: MiamiRoute [-nqv] kommando modifierare argument

Argument:

-n

Försök inte visa värnamn och nätverksnamn symboliskt när händelser rapporteras. (Att omvandla mellan symboliska namn och numeriska ekvivalenter kan vara ganska lång tid och kan kräva att nätverket fungerar korrekt; det kan alltså vara lämpligt att inte göra detta, speciellt vid försök att reparera nätverk.)

-q

Visar inte någon utmatning.

-v

Visar utförlig utmatning. Fler detaljer visas.

Kommandon:

add

Lägger till en väg.

flush

Tar bort alla vägar. Var mycket försiktig när du använder det här kommandot. Det tar också bort några av Miamis normala vägar. Såvida du inte vill reparera dem manuellt efteråt måste du starta om Miami för att återfå normal funktion.

delete

Tar bort en viss väg.

change

Ändrar aspekter på en väg (såsom dess brygga).

get

Slår upp och visar vägen till en destination.

monitor

Visar kontinuerligt eventuella ändringar på routningsinformation, uppslagningsmissar eller misstänkta nätverksuppdelningar. OBS: utan en implementation av "routed" är inte det här kommandot så väldigt användbart.

Kommandot MiamiRoute behövs normalt inte med protokollstackar med bara ett gränssnitt som Miami, och är mycket komplext och svårt att använda. För en utförligare diskussion, se BSD-dokumentationen för kommandot "route".

Den enda egentligen användbara tillämpningen av MiamiRoute för tillfället är att undersöka vägar till värdar, till exempel för att ta reda på tur och returtider eller MTU-värden för vägen. För att göra detta, använd syntaxen:

```
MiamiRoute get hostname
```

För att undersöka den kompletta routningstabellen, använd kommandot "MiamiNetStat -r", inte MiamiRoute.

1.138 Miami.guide/NODE_UTILITY_SYSCTL

MiamiSysCtl

=====

Undersöker och ändrar några av Miamis interna variabler.

Användning:

MiamiSysCtl [-n] variabel

Undersöker värdet på en variabel.

MiamiSysCtl [-n] -w variabel=värde

Ändrar värdet på en variabel.

MiamiSysCtl [-n] -a

Visar en lista över alla variabler.

MiamiSysCtl [-n] -A

Visar en lista över alla variabler plus ytterligare information om domäner som inte kan nås med MiamiSysCtl.

Argument:

-n

Visar bara numerisk utmatning.

Förklaring av alla variabler:

net.inet.ip.forwarding/redirect

De här variablerna har ingen funktion för en stack med ett gränssnitt som Miami.

net.inet.ip.ttl

Det förvalda ttl-värdet ("tid att leva") för paket som Miami skickar. Bör vara det förvalda värdet 64.

net.inet.ip.rtxpire/rtminexpire/rtmaxcache

Tider och storlek för vägkloning. Du bör inte ändra dessa värden.

net.inet.op.sourceroute

Kontrollerar hur paket som innehåller en IP-källväg ska hanteras. Den här variabeln är bara viktig för routrar.

net.inet.op.pathmtudisc

Anger om det ska vara möjligt att upptäcka vägens MTU-värde. Förvalt är att det ska vara möjligt, men om du är uppkopplad genom gamla, felaktiga routrar och har problem med TCP-trafiken kan du försöka med att stänga av den här funktionen.

net.inet.icmp.maskrepl

Anger om Miami skickar nätmasken i svaret på ICMP-frågor om masken. Om nätmasken är rätt konfigurerad och du aktiverar den här funktionen kommer alla andra datorer på det lokala nätverket som kör MiamiInitSANA2 eller Miami att kunna hitta

rätt nätmask via ICMP.

net.inet.tcp.rfc1323

Aktiverar TCP-extensions enligt RFC1323. De här extensionerna kan inte användas med PPP/SLIP-VJC, varför de bara bör vara aktiverade för bussgränssnitt, inte "punkt till punkt"-gränssnitt.

net.inet.tcp.rfc1644

Aktiverar T/TCP.

net.inet.tcp.mssdfmt

Sätter den förvalda maximala segmentstorleken för TCP. Normalt bör detta värde inte ändras. Det används normalt inte, eftersom Miami använder vägens MTU-värde för att avgöra optimala värden för den maximala segmentstorleken.

net.inet.tcp.rttdeflt

Kontrollerar TCPs omsändningstider och bör inte ändras.

net.inet.tcp.keepidle/keepintvl

Kontrollerar TCPs "keep alive"-tider och bör inte ändras.

net.inet.tcp.sendspace/recvspace

Kontrollerar de förvalda fönsterstorlekarna för sändning och mottagning och bör normalt inte ändras.

net.inet.tcp.bulkftp

Reserverad för framtida användning. Gör för tillfället inte någonting. Ändra den inte.

net.inet.tcp.initwin

Anger antalet paket i det initiala TCP-fönstret för nya uppkopplingar. Förvalt är 1, men det har nyligen visats att i vissa fall kan 2 eller 3 ge bättre prestanda.

net.inet.tcp.fastlocal

Aktiverar en optimering som ger mycket snabbare uppkopplingar mot den lokala värden.

net.inet.udp.checksum

Aktiverar UDP-kontrollsummor för alla utåtgående paket. Bör alltid vara aktiverad.

net.inet.udp.maxdgram/recvspace

Kontrollerar tröskelvärden för UDP-paket och bör inte ändras.

dns.cache.size

Kontrollerar storleken på Miamis interna DNS-cache.

dns.cache.flush

Sätt den här variabeln till 1 för att tömma Miamis interna DNS-cache.

inetd.retrytime/toomany/cntintvl/maxbuiltin

Kontrollerar InetDs inbyggda skydd mot SYN-floder. Om du kör en mycket upptagen webbserver kan du behöva justera dessa variabler

(speciellt "toomany") för att klienter inte ska få felmeddelanden i onödan om din dator är överbelastad.

dns.cache.enabled

Kontrollerar Miamis inbyggda DNS-cache. Det förvalda värdet är 2, vilket betyder att alla värdposter lagras i cachen. Om värdet är 1 kommer bara värdar med en enda IP-adress att lagras, vilket gör att problem med värdar som har roterande IP-adresser undviks. Om värdet är 0 kommer Miamis DNS-cache att vara helt avstängd. Du bör bara stänga av cachen om du har en mycket snabb uppkoppling till en lokal DNS-server.

dns.cache.split

Den här variabeln är normalt 0, vilket indikerar att Miami använder en gemensam cache för normala och omvända sökningar. Om värdet är 1 använder Miami separata cachar för normala och omvända sökningar. Detta gör diagnosutmatning (från till exempel MiamiNetStat) långsammare, men garanterar "korrekt" omvänd sökning av alla IP-adresser (genom att använda PTR-sökningar).

inetd.retrytime

Kontrollerar efter hur lång tid InetD försöker återskapa en anslutning om det första försöket misslyckades.

inetd.toomany

Anger det maximala antalet uppkopplingar InetD accepterar inom ett givet tidsintervall.

inetd.cntintvl

Anger tidsintervallet för inetd.toomany.

inetd.maxbuiltin

Anger det maximala antalet inbyggda servrar som startas av InetD.

inetd.processpri

Anger processprioriteten för servrar som startas av InetD. Förvalt är -5. Du bör öka detta värde om du kör några CPU-intensiva bakgrundsprocesser (till exempel klienten för RC5-utmaningen). Annars kanske dina servrar inte får någon CPU-tid.

inetd.diagbufsize

Anger storleken på anslutningsbuffrar för diagnostikrelaterade InetD-tjänster ("chargen", "echo" med flera). Det förvalda värdet är 4096, alltså mindre än de normala buffrarna för UDP/TCP-kontakter. för att minska påverkan av försök att överbelasta systemet.

inetd.diagtimeout

Tid i sekunder efter vilken uppkopplingar till diagnostikrelaterade InetD-tjänster avslutas.

socket.maxqlen

Anger längden på anslutningskön för den femte listen()-parametern. Förvalt är 7, men om du är uppkopplad mot ett mycket snabbt nätverk och har tillräckligt med minne kanske

du vill öka detta värde för att minska effekterna av SYN-flodattacker.

1.139 Miami.guide/NODE_UTILITY_TCPDUMP

MiamiTCPDump
=====

Dumpar trafik på ett nätverk efter att ha filtrerat det.

Användning: MiamiTCPDump [-adflnNOqStvx] [-c antal] [-F fil] [-i gränssnitt] [-r fil] [-s längd] [-T typ] [-w fil] [uttryck]

Argument:

- A
Används tillsammans med "-x": visar paketen i textformat såväl som hexformat.
 - a
Försöker konvertera nätverks- och broadcastadresser till namn.
 - c antal
Avslutar efter att ha mottagit <antal> paket.
 - d
Dumpar den kompilerade paketfiltreringskoden, i en för människor läsbar form, till standardutmatningen och avslutar.
 - dd
Dumpar den kompilerade paketfiltreringskoden som ett programfragment.
 - ddd
Dumpar den kompilerade paketfiltreringskoden som decimala värden (föregångna av en räknare).
 - f
Visar "främmande" Internetadresser numeriskt istället för symboliskt.
 - F fil
Använder <fil> som indata för filtreringsuttrycket. Ett eventuellt uttryck på kommandoraden ignoreras.
 - i gränssnitt
Lyssnar på <gränssnitt> (för tillfället "lo0" eller "mi0"). Om argumentet inte anges söker MiamiTCPDump igenom systemets gränssnittslista efter det lägst numrerade och konfigurerade gränssnitt som är uppe (exklusive lokala loopar). Om flera gränssnitt skulle vara aktuella används det som hittas först. För tillfället är resultatet av den här sökningen alltid "mi0".
 - l
-

Gör standardutmatningen radbuffrad. Användbart om du vill se datan medan den fångas.

-n

Konverterar inte adresser (alltså värdadresser, portnummer med mera) till namn.

-N

Visar inte domännamnskvalificerare för värdnamn. Till exempel kommer MiamiTCPDump att visa "nic" istället för "nic.ddn.mil".

-O

Kör inte optimeraren för paketfiltreringskoden. Detta är bara användbart om du misstänker att det finns ett fel i optimeraren.

-P

Använder inte promiskuöst läge. Om en MNI-drivrutin används kommer MiamiTCPDump normalt att sätta gränssnittet i promiskuöst läge så länge som MiamiTCPDump kör. Använd det här argumentet för att stänga av den funktionen, alltså att låta gränssnittet vara i sitt normala läge.

-q

Snabb (tyst?) utmatning. Visar mindre protokollinformation så att utdatoraderna blir kortare.

-s längd

Fångar <längd> databyte från varje paket istället för det förvalda antalet av 68. 68 byte är tillräckligt för IP, ICMP, TCP och UDP, men kan trunkera protokollinformation från namnservrar och NFS-paket (se nedan). Paket som trunkeras på grund av en begränsad längd indikeras i utmatningen med "[proto]", där <proto> är namnet på den protokollnivå på vilken trunkeringen inträffade. Observera att längre paketfångster både ökar tiden det tar att behandla paket och minskar antalet paket som buffras. Detta kan leda till att paket tappas. Du bör begränsa <längd> till det minsta värde som fångar den protokollinformation du är intresserad av.

-S

Visar absoluta, istället för relativa, TCP-sekvensnummer.

-T typ

Tvingar paket som väljs av <uttryck> att tolkas som den angivna typen. Typer som för tillfället kan användas är:

- * rpc (Remote Procedure Call)
- * rtp (Real-Time Applications protocol)
- * rtcp (Real-Time Applications control protocol),
- * vat (Visual Audio Tool),
- * wb (distributed White Board).

-t

Visar inte en tidsstämpel på varje dumprad.

-tt

Visar en oformaterad tidsstämpel på varje dumprad.

-v

(Något) utförligare utmatning. Till exempel visas tid att leva och typen av serviceinformation för ett IP-paket.

-vv

Ännu utförligare utmatning. Till exempel visas fler fält för NFS-svarspaket.

-w fil

Skriver de otolkade paketen till <fil> istället för att tolka och visa dem på standardutmatningen. De kan senare visas med argumentet "-r". Standardutmatningen används om <fil> är "-".

-x

Visar varje paket (utom dess länknivåhuvud) i hexformat. Upp till det antal byte som angetts i argumentet "-s" visas.

<uttryck> väljer vilka paket som ska dumpas. Om inget <uttryck> anges kommer alla paket på nätet att dumpas. Annars kommer bara paket för vilka <uttryck> är sant att dumpas.

Syntaxen för <uttryck> är extremt omfattande och utanför den här manualens område. För en komplett beskrivning av syntaxen och alla detaljer om utmatningsformatet, se dokumentationen för den fritt distribuerbara BSD-versionen av "tcpdump". Här är några exempel på korrekta uttryck:

```
"host sundown"
```

Visar alla paket som kommer till eller skickas från "sundown".

```
"host helios and ( hot or ace )"
```

Visar trafiken mellan "helios" och antingen "hot" eller "ace".

```
"ip host ace and not helios"
```

Visar alla IP-paket mellan "ace" och alla värdar utom "helios".

```
"tcp[13] & 3 != 0"
```

Visar start- och slutpaket (SYN och FIN) för varje TCP-konversation.

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

Visar alla ICMP-paket som inte är "echo"-förfrågningar eller -svar (alltså inte pingpaket).

1.140 Miami.guide/NODE_UTILITY_TRACEROUTE

MiamiTraceRoute

=====

Visar den väg paket tar till en nätverksvärd.

Användning: MiamiTraceRoute [-m max_ttl] [-n] [-p port] [-q antal_frågor] [-r] [-s källadr] [-t tat] [-v] [-w väntetid] värd [paketstorlek]

Argument:

- m max_ttl
Sätter maximal tid att leva (maximalt antal hopp) för de utgående paketen. Förvalt är 30 hopp.
 - n
Visar hoppadresser numeriskt istället för symboliskt och numeriskt (undviker en DNS-sökning från adress till namn för varje brygga som hittas på vägen).
 - p port
Sätter grund-UDP-portnummer för skickade paket (förvalt är 33434). MiamiTraceRoute hoppas att inget lyssnar på grund-UDP-porten <port>+antal_hopp-1 vid slutdestinationen (så ett ICMP-meddelande av typen PORT_UNREACHABLE kommer att returneras för att avsluta spårningen). Om något lyssnar på en port i det förvalda intervallet kan det här argumentet användas för att välja ett portintervall som inte används.
 - q antal_frågor
Sätter antal paket per "ttl" till <antal_frågor> (förvalt är 3 paket).
 - r
Använder inte de normala routningstabellerna och skicka paketen direkt till en värd på ett anslutet nätverk. Om värden inte är på ett direkt anslutet nätverk returneras ett fel.
 - s källadr
Använder <källadr> (som måste anges som en IP-adress, inte som ett värddamn) som källadressen i de utgående paketen. På värddar med mer än en IP-adress kan det här argumentet användas för att tvinga källadressen att vara något annat än IP-adressen för det gränssnitt som paketen skickas på. Om IP-adressen inte är ett av den här datorns gränssnittsadresser returneras ett fel och inget skickas.
 - t tat
Sätter typ av tjänst i paketen till värdet <tat> (förvalt är 0). Värdet måste vara ett heltal i intervallet 0 - 255. Det här argumentet kan användas för att se om olika tjänstetyper resulterar i olika vägar.
 - v
Utförlig utmatning. Mottagna ICMP-paket utöver TIME_EXCEEDED och UNREACHABLE visas.
 - w väntetid
Sätter tid (i sekunder) att vänta på ett svar på ett paket (förvalt är 3 sekunder).
-

1.141 Miami.guide/NODE_COMPATIBILITY

Kompatibilitet

Hittills har Miami fungerat med alla AmiTCP-klienter och servrar som det har testats med, med ett undantag:

Versionen av "telnet" som kommer med AmiTCP 4.x fungerar normalt inte med Miami. Det beror på att denna version av "telnet" använder sig av odokumenterade funktioner i "TCP:" som inte kan emuleras av Miami.

Det finns tre lösningar på det här problemet:

- * Uppgradera till Miami Deluxe som innehåller "MiamiTelnet", en ny, fullt utrustad telnetklient för AmigaOS.
- * Använd en annan version av telnet, till exempel "AmTelnet", en trevlig MUI-baserad grafisk telnetklient som finns tillgänglig från www.vapor.com, telnetversionen som finns tillgänglig från Aminet i `comm/tcp`, ett terminalprogram tillsammans med `telser.device` eller "napsaterm" i telnetläge.
- * Installera den version av "inet-handler" som följer med AmiTCP 4.0demo, skapa en lämplig post i Mountlist för "TCP:" och skriv "mount TCP:" innan Miami startas. "telnet" kommer sedan att använda AmiTCP-versionen av "TCP:", som givetvis fortfarande använder sig av Miamis TCP/IP-stack, istället för den version av "TCP:" som är inbyggd i Miami.

1.142 Miami.guide/NODE_RESTRICTIONS

Begränsningar

Demoversionen har följande begränsningar:

- * Efter sextion minuter lägger modemmet på. SANA-II-uppkopplingar avbryts efter trettio minuter.
 - * Det är inte möjligt att hålla TCP-uppkopplingar vid liv när modemmet lägger på.
 - * Automatisk uppkoppling ("autopkoppl.") efter passiv nedkoppling och exekvering av ARExx- och shellskript på sidan "Händelser" är ej tillgängliga.
 - * Antalet telefonnummer i uppringaren är begränsat till tre.
-

- * Telefonloggning är avstängd.
- * Valen för GUI-förändringar är avstängda.
- * Multicast och T/TCP fungerar inte.
- * IP-filtret är inte tillgängligt.
- * Skydd mot pingfloder är inte tillgängligt.
- * Sortering, sammanslagning och import/exportfunktionerna för urklippet på sidan "Databas" finns ej tillgängligt.
- * PPP-motringning är inte tillgängligt.
- * Återanropsfunktioner för paketövervakning, för externa paketmonitorer som MiamiTCPDump, fungerar inte.
- * Händelser i systemloggen kan inte exporteras till syslog.library.
- * Nyttoprogrammen MiamiIfConfig, MiamiRemind, MiamiRoute, MiamiSysCtl, MiamiTCPDump, alla multicastverktyg samt biblioteken miamibpf.library och miamipcap.library kan inte användas.
- * Stöd för MS-CHAP är inte tillgängligt.

GUI-motorn GTLayout har några få begränsningar jämfört med MUI-versionen:

- * Sortering genom drag och släpp i listor är inte tillgängligt.

1.143 Miami.guide/NODE_HISTORY

Historia

Version 3.2

Släppt version

- * Åtskilliga ändringar. Se filen CHANGES i Miamis huvudarkiv.

Version 3.0

Släppt version

- * För många ändringar för att räknas upp här. Många delar av programmet har skrivits om från början. Nyckelfiler för v3 krävs nu.

Version 2.1p

Släppt revision

- * Den senast officiella versionen före 3.0. Mest rättade fel och små ändringar sedan tidigare versioner.

1.144 Miami.guide/NODE_FUTURE

I framtiden

Mina omedelbara planer för de kommande versionerna av Miami är:

- * Ett nytt API för att klienter automatiskt ska kunna hantera upp- och nedkopplingar.
- * Ett helt nytt API ("ANDI") för mycket enklare, protokolloberoende tillgång till TCP/IP-funktioner från applikationer.

ISDN är en annan stor fråga. Externa ISDN-terminaladapttrar stöds redan till fullo. Det gör även ISDN-Master-kortet i de flesta lägen, förutom i synkron PPP i HDLC-läge. Det läget kräver nya drivrutiner och ett nytt API, men folk arbetar på detta...

Den enda riktigt stora (avsiktliga) begränsningen med Miami är att det är begränsat till ett gränssnitt. Ytterligare en protokollstack kallad "Miami Deluxe" är planerad till senare under 1999. För närvarande innehåller mina planer för Miami Deluxe stöd för flera gränssnitt och antagligen många andra användbara funktioner för routrar, såsom stöd för SOCKS-demoner, IP-maskering och brandväggsfunktioner.

Det kommer att finnas en rabatt för registrerade användare av Miami som vill registrera Miami Deluxe.

1.145 Miami.guide/NODE_SUPPORT

Support

Det finns flera sätt att få teknisk support, uppdateringar, med mera:

e-post

kruse@nordicglobal.com

vanlig post

Nordic Global Inc.
Attn: Holger Kruse
PO Box 780248
Orlando FL 32878-0248
USA

WWW

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

e-postlistor

Skriv "SUBSCRIBE miami-talk-ml" eller "SUBSCRIBE miami-announce-ml" i brevkroppen och skicka brevet till

"Majordomo@nordicglobal.com".

1.146 Miami.guide/NODE_ACKNOWLEDGEMENTS

Tack

Stort tack går till:

- * De tidiga alpha- och betatestarna Karl Bellve, Mike Fitzgerald, Adam Hough, Daniel Saxer, Stefan Stuntz och Oliver Wagner.
 - * Karl Bellve och Daniel Saxer för deras stora stöd.
 - * NSDi för det första fritt tillgängliga TCP/IP-protokollet för AmigaOS och för dess användbara API.
 - * James Cooper, Steve Krueger och Doug Walker för utvecklingsmiljön SAS/C och för deras stora stöd.
 - * Stefan Stuntz för hans trevliga grafiska användargränssnitt MUI.
 - * Klaus Melchior för hans MUI-klass "Busy.mcc".
 - * Robert Reiswig för att han lånade ut viktig datorutrustning.
 - * University of California för deras fortsatta framgångsrika arbete på den utmärkta nätverkskällkoden till BSD.
 - * Reinhard Spisser och Sebastiano Vigna för deras Amigaport av "makeinfo".
 - * Paul Trauth, vinnaren i Miamis logotyp tävling, för hans trevliga bildsamling.
 - * John Pszeniczny för hans trevliga varianter av logotypen till "Miami".
 - * Jim Szutowicz för hans flerfärgsversioner av logotypen till "Miami".
 - * Martin Huttenloher och Stefan Stuntz för deras tillstånd att använda MagicWB-bilder i Miami.
 - * Roman Patzner för hans nya ikondeSIGN.
 - * Olaf Barthel för gtlayout.library och hans hjälp med att finna en del problem.
 - * Alla användare som bestämmer sig för att registrera Miami.
-