

ProNET

COLLABORATORS

	<i>TITLE :</i> ProNET		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		December 8, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ProNET	1
1.1	ProNET Anleitung	1
1.2	ProNET Einleitung	2
1.3	ProNET Systemanforderungen	2
1.4	ProNET Herstellung des Parallelkabels	2
1.5	ProNET Das AmigaDOS	3
1.6	ProNET Installation	4
1.7	ProNET Installation - pronet.device	5
1.8	ProNET Installation - File System	6
1.9	ProNET Installation - Utilities	8
1.10	ProNET Start des Netzwerks	8
1.11	ProNET Beispielkonfiguration	9
1.12	ProNET pronet-talk	10
1.13	ProNET pronet-run	11
1.14	ProNET pronet-page	11
1.15	ProNET Konfiguration	11
1.16	ProNET Bekannte Probleme	12
1.17	ProNET Update von V1, Bemerkungen	12
1.18	ProNET Registration	13
1.19	ProNET Support	13
1.20	ProNET Credits & Danksagungen	14
1.21	ProNET Geschichte	14
1.22	ProNET Zukunft	15
1.23	ProNET Frequently Asked Questions	15
1.24	ProNET Disclaimer	17
1.25	ProNET License	17
1.26	ProNET Sourcecode	18
1.27	ProNET Programmers	19
1.28	ProNET Unterstützte DosPackets	20

Chapter 1

ProNET

1.1 ProNET Anleitung

Dies ist die Anleitung zu
ProNET V2

Ein Netzwerksystem

(c) Copyright 1994-1995 by Michael Krause

++ ProNET V2 ist (extrem günstige, nicht-verkrüppelte) SHAREWARE ++

Einleitung	Was ist ProNET
Systemanforderungen	Was brauche ich, um ProNET zu benutzen
Installation	Wie wird ProNET installiert
Start des Netzwerks	Wie man das Netz startet
Beispielkonfiguration	Eine komplette Beispielkonfiguration
pronet-talk	Wie man sich übers Netz unterhält
pronet-run	Programme auf anderen Rechnern starten
pronet-page	Nachrichten senden
Konfiguration	Was kann man ändern
Unterstützte DosPackets	Unterstützte DosPackets
Bekannte Probleme	Probleme beim Betrieb
Update von V1, Bemerkungen	Für Ex-User von Version 1
Sourcecode	Die Sources
Programmierer	Neue Treiber programmieren
Registration	Wie registriert man sich
Support	Neue Versionen, Bug-Reports
Credits & Danksagungen	Wer hat zu ProNET beigetragen
Geschichte	Wie ProNET entstand
Zukunft	Was als nächstes getan wird
FAQ	Häufige Fragen

Disclaimer	Garantie???
License	Verbreitungsvorschriften

1.2 ProNET Einleitung

ProNET ist eine Hardware-/Software-Kombination, die zwischen zwei oder auch mehr Amigas ein mehr oder weniger einfaches Netzwerk installiert. Das bedeutet, Sie können alle Geräte wie Festplatten, Floppies oder CD-ROMs auf allen Rechnern im Netz benutzen, als wenn sie im benutzenden Rechner eingebaut wären. Dazu können Sie serielle (Null-Modem) oder parallele Kabel benutzen.

Wenn Sie früher ParNet benutzt haben sollten, sind sie mit dieser Art von Software bestimmt bekannt, aber ProNET hat ein paar Vorteile:

- richtet ein neues AmigaDOS-Device für jedes Ziel ein (kein NET:)
- erkennt Diskwechsel
- unterstützt die meisten V40-DosPackets
- 'cd's in Netzwerkverzeichnisse funktionieren!
- geschrieben in Assembler, deswegen kürzer und auch schneller

Größere Veränderungen zu Version 1 sind z.B.:

- komplett neues pronet.device, modulares Treiberkonzept
- kein Freezen während des Datentransfers
- Klienten können jetzt jederzeit rebooten
- seriellles Treibermodul
- ReqTools-Requester erkennen auch ProNET-Laufwerke
- viele Bug-Fixes
- kompletter Sourcecode mitgeliefert
- völlig neue Dokumentation

1.3 ProNET Systemanforderungen

Die gesamte Software läuft auf allen Systemen mit mindestens Kickstart 1.3 (Kick 1.2 wurde nicht getestet, müßte allerdings auch funktionieren); eine Ausnahme bildet der Handler, der Kickstart 2.0 benötigt!

Die Hardware, die aus einem einfachen Kabel besteht, wird an den Parallelport angeschlossen. Sie können Ihr altes ParNet-Kabel weiterbenutzen -- falls sie ParNet nie benutzt haben, hier ist die Anleitung:

1.4 ProNET Herstellung des Parallelkabels

```
{ Copied from the original ParNet distribution }  
  { Translated by Michael Krause }
```

Das einfachste wäre es, ein fertiges 25er-Kabel zu kaufen und nur zu modifizieren, wie folgt beschrieben. Damit das Kabel passt, sollten Sie

nachsehen, wie Ihre Parallel-Buchse aussieht - Bei allen Amiga-Modellen außer dem A1000 sollte dies eine weibliche sein, d.h. es ist eine Buchse und kein Stecker !!

VERBINDEN SIE NIEMALS DIE COMPUTER MIT EINEM UNMODIFIZIERTEN KABEL !!

PRÜFEN SIE DAS KABEL LIEBER DREIMAL VORHER, ALS SICH HINTERHER DREIMAL ÜBER EINE KAPUTTE KISTE ZU ÄRGERN !!

Verbindungen: D7-D0, SEL, POUT und BUSY beider Computer verbinden
ACK (FLAG-Interrupt) und SEL am Stecker zusammenschließen.

(2-9)	D7-D0	-----	D7-D0	
(12)	POUT	-----	POUT	
(11)	BUSY	-----	BUSY	PARALLEL PORT
(13)	SEL	--+-----+--	SEL	
(10)	ACK	-/ \-	ACK	
(18-22)	GND	-----	GND	(18-22)

WARNUNG! WECHSELWIRKUNGEN MIT DEM SERIELLEN PORT !!

Die RI-(Ring Indicator-)Leitung am _seriellen_ Anschluß Ihres Amigas benutzt die SEL-Leitung, um einen Transistor zu speisen. Daraus ergeben sich aber leider einige Probleme, da diese Leitung am Netzwerk beteiligt ist.

Stellen Sie also sicher, daß entweder:

- (a) kein seriell Kabel angeschlossen ist, oder
- (b) daß es die RI-Leitung nicht benutzt, oder
- (c) daß Ihr Modem die RI-Leitung intern nicht benutzt !

Bemerkung hierzu vom ProNET-Autor: Ich kenne niemanden, der Probleme damit hatte, ParNet und ein Modem gleichzeitig zu benutzen - allerdings heißt das nicht, daß ich irgendeine Verantwortung übernehme, falls Schäden entstehen !!

1.5 ProNET Das AmigaDOS

In diesem Teil versuche ich, Begriffe wie 'Device', 'Handler' und 'File System' zu erläutern. Wenn Sie finden, daß Sie hierüber genug wissen, können Sie diese Sektion überspringen und gleich mit dem Installieren beginnen. Wenn Sie den Installationsvorgang nicht verstehen, kann Ihnen diese Sektion vielleicht weiterhelfen!

Weil AmigaDOS mehr als nur ein Speichermedium verwalten kann, bekommt jedes seinen eigenen Namen. Die prominentesten Beispiele wären da 'DF0:' für die eingebaute Floppy und 'HD0:' für die erste Festplattenpartition. Immer, wenn Sie auf eine Datei auf so einem Medium zugreifen, geben Sie diesen Namen an, es sei denn natürlich, dieses Medium ist das Wurzelverzeichnis des aktuellen Verzeichnisses.

All diese Namen beschreiben etwas von AmigaDOS, das man AmigaDOS-Device oder -Gerät nennt. 'CD0:', 'HD7:', 'PRT:' und 'RAM:' beschreiben alle verschiedenen AmigaDOS-Geräte, oder auch einfach nur Devices. Wie Sie vielleicht am Beispiel 'RAM:' bemerkt haben, ist das AmigaDOS-Device nur ein Symbol für ein Speichermedium, denn die Ram Disk ist schließlich kein reeller Teil der Hardware.

Zu jedem Device gehört ein Kontrollprogramm, das man Handler oder FileSystem nennt. Der Unterschied zwischen beiden ist, daß ein FileSystem Dateien (Files) auf einem Medium verwaltet, während ein Handler bei Devices benutzt wird, die nur Daten umherschreiben, wie z.B. der Drucker 'PRT:'. Oder konnten Sie schon einmal ein Directory auf dem Drucker erstellen?

Sobald ein FileSystem den Auftrag erhält, ein bestimmtes File zu lesen oder zu schreiben, reicht es einige Daten an eine weitere Kontrollebene weiter, nämlich an ein Exec-Device. Dieses kontrolliert nun endlich die Hardware. Warum der Umweg? Vergleichen Sie doch mal eine Festplatte mit einem Diskettenlaufwerk. Beide sind sehr ähnlich aufgebaut; sie bestehen aus Zylindern, Tracks und Sektoren, nur die Größe unterscheidet sich. Deswegen werden beide Medien auch von nur einem FileSystem kontrolliert -- die kleinen Unterschiede, nämlich die unterschiedliche Ansteuerung der Hardware (Schreib-/Leseköpfe und Motoren), werden von je einem Exec-Device gemeistert. Durch dieses Verfahren spart man eine Menge Speicher, denn beide Medien teilen sich ein großes Stück Programmcode. Beispiele für Exec-Devices sind z.B. das 'trackdisk.device' oder das 'scsi.device'.

Wenn man sich übrigens mal Handler ansieht, stellt man fest, daß bei einigen das Exec-Device fehlt, oder diese gar keine Motoren steuern, sondern nur Schnittstellen, wie z.B. das 'serial.device'.

Um ein neues AmigaDOS-Device anzumelden, benutzt man den Mount-Befehl, der einen sogenannten MountList-Eintrag benötigt. Es gibt zwei Orte, wo man diesen ablegen kann: Der vorgegebene Weg vor Kickstart 2.0 war, alle Einträge in der Datei 'DEVS:MountList' zu sammeln. Ab Kickstart 2.0 kann man auch für jedes Device eine eigene Datei erzeugen und diese nach 'SYS:Storage/DOSDrivers' kopieren. Das File sollte dann genauso genannt werden, wie das Device, das es beschreibt. Schauen Sie mal in das Verzeichnis hinein, Sie werden dort schon einige Beispiele anfinden.

Das Mount-Kommando wird mit dem anzumeldenden Devicenamen aufgerufen und durchsucht die oben beschriebenen Orte nach dem MountList-Eintrag. Zusätzlich werden bei jedem Systemstart automatisch alle Devices, deren Einträge in 'DEVS:DosDrivers' stehen, gemountet.

Bitte bringen Sie nicht die Begriffe 'Exec Device' und 'AmigaDOS Device' durcheinander, obwohl beide i.A. bloß 'Device' oder 'Gerät' genannt werden.

1.6 ProNET Installation

- Wie installiert man ProNET -

Das Parallelkabel
AmigaDOS

Das Verbindungskabel
Immer gut zu wissen

pronet.device	Die Übertragungsroutinen
File System	Datei-Netzwerk
Utilities	Andere Tools

1.7 ProNET Installation - pronet.device

Kopieren der Dateien

Das `pronet.device` ist der Kern des gesamten ProNET-Pakets. Es enthält alle Routinen, die den Datentransfer zwischen mehreren Amigas verwalten. Es enthält KEINE eigentlichen Übertragungsroutinen, wie es der Fall in Version 1 war, weil es sich als sehr schwierig herausstellte, neue Übertragungsroutinen für andere Schnittstellen zu schreiben. Dieses Device sollte ins `'DEVS:'`-Verzeichnis aller am Netz teilnehmenden Maschinen kopiert werden.

Die Übertragungsroutinen sind externe Programmmodule, die im `'DEVS:ProNET'`-Verzeichnis gelagert werden. Bitte kopieren Sie alle mitgelieferten Module dorthin. Wichtig: Nicht direkt nach `'DEVS:'` kopieren, sondern in ein neues ProNET-Verzeichnis!

Bis jetzt gibt es die Module `'internal-parallel'` für den internen Parallelport (wirklich!), wie es die einzige Möglichkeit in V1 war, sowie `'serial'` für jeden beliebigen Port, auf den über ein `'serial.device'`-kompatibles Device zugegriffen wird, z.B. `'BaudBandit.device'` oder `'gvpserial.device'`.

Erstellen des Konfigurationsfiles

Erstellen (oder kopieren Sie das mitgelieferte Beispiel) Sie jetzt bitte die Datei `'DEVS:ProNET.config'`. Diese wird Konfigurationsdaten für alle Programme enthalten, die das `pronet.device` benutzen.

Sie MUß mindestens eine Zeile der folgenden Form enthalten:

```
pronet-device-<Unitnummer>: <Treibername> <Treiberdaten>
```

Eine solche Zeile beschreibt eine sogenannte Unit des Devices, die einer beliebigen, von einem ProNET-Treiber ansprechbaren Schnittstelle entspricht. Je mehr Ports und Treiber Sie haben, desto mehr Units können Sie installieren. Sie werden die Möglichkeit haben, jede Unit zur Verbindung mit anderen Amigas zu benutzen. Wenn Sie z.B. eine Multiport-Karte mit 3 seriellen Ports haben, könnten Sie sich später mit 5 Amigas vernetzen, indem Sie die Multiport-Karte zusammen mit internem Parallel- und Seriellport benutzen.

Jede Unit wird durch eine beliebige positive Zahl gekennzeichnet (`<Unitnummer>`). `<Treibername>` ersetzen Sie bitte durch den Namen des externen Treibermoduls, `<Treiberdaten>` sind Angaben, die der jeweilige Treiber zum korrekten Funktionieren benötigt. Diese können sich natürlich von Treiber zu Treiber unterscheiden.

Installieren des 'internal-parallel'-Treibers

Als erstes werden wir diesen Treiber als Unit Nummer 0 einrichten. Natürlich können Sie auch eine beliebige andere Zahl benutzen, es ist Ihre Entscheidung! Sie müssen diesen Treiber sowieso nicht benutzen.

Wir erstellen die (oder ändern die bereits vorhandene) Zeile:

```
pronet-device-0: internal-parallel 0 5
```

Die Zahl 5 wird vom Treiber benötigt und gibt die Priorität des Transferprozesses an. Wenn Sie nicht wissen, wovon ich rede, lassen Sie diese Zahl einfach stehen!

Die Zahl 0 ist die sogenannte Rechnernummer und ist von großer Bedeutung für das korrekte Funktionieren des Treibers. Zwei Rechner, die über den 'internal-parallel'-Treiber vernetzt werden, müssen zwei verschiedene Nummern hier stehen haben: Eine Maschine '0', die andere '1'. Wenn Sie beiden Seiten die gleiche Nummer geben, riskieren Sie gefährliche Netzwerk-Ausfälle (Nein, Sie machen Ihren Rechner nicht kaputt, es funktioniert nur nicht so, wie Sie es wollen).

Dieser Treiber benötigt ein ParNet-kompatibles Kabel am eingebauten Parallelport.

Installieren des 'serial'-Treibers

Hier müssen wir natürlich eine andere Unit-Nummer benutzen als beim 'internal-parallel'-Treiber, z.B. '1':

```
pronet-device-1: serial BaudBandit.device 0 38400
```

'BaudBandit.device' ist der Name des 'serial.device'-kompatiblen (Exec-)Devices, das Sie benutzen wollen. Falls Sie dieses nicht besitzen, können Sie natürlich auch das 'serial.device' selbst benutzen. '0' beschreibt die Serial-Unit (nicht verwechseln mit der ProNET-Unit); auf Multiport-Boards würde hiermit die Nummer der Schnittstelle festgelegt werden. Außerdem gibt 38400 noch die BPS-Rate an, mit der die Schnittstelle betrieben werden soll. Sie sollte natürlich auf beiden Seiten gleich sein -- probieren Sie, wie hoch Sie gehen können!

Der 'serial'-Treiber benötigt ein 7-Draht/RS232 NullModem-Kabel an der betroffenen Schnittstelle.

1.8 ProNET Installation - File System

Ich werde den Vorgang hier nur aus der Sicht einer Hauptmaschine beschreiben. Sie können das Netzwerk natürlich auch 'über Kreuz' betreiben.

Kopieren der Files

Kopieren Sie zuerst den pronet-server in das 'C:'-Directory oder irgendwo anders in den Pfad des Rechners, _von dem_ sie Devices importieren wollen.

Kopieren Sie den pronet-handler in das 'L:'-Directory des Hauptrechners.

Erstellen von MountList-Einträgen

Sie müssen für jedes zu importierende Device einen neuen MountList-Eintrag erstellen. Ich werde den Vorgang hier nur für ein Beispiel beschreiben. Nehmen wir an, der Server-Rechner hat eine Festplatte namens 'HD0:', die wir auf unserem Hauptrechner mitbenutzen wollen. Als erstes müssen wir uns über einen neuen Namen klar werden. Da unser Hauptrechner ebenfalls über eine Festplatte 'HD0:' verfügt, können wir die neue wohl kaum genauso nennen. Wir benutzen deswegen den Namen 'HD4:'.

Was wir im folgenden also erreichen wollen, ist das Erstellen eines neuen AmigaDOS-Devices mit dem Namen 'HD4:', das eine exakte Kopie der 'HD0:' des anderen Rechners ist.

Laden Sie jetzt Ihren Lieblingseditor und öffnen Sie 'DEVS:MountList'. Gehen Sie ans Ende und fügen Sie die ProNET-Beispiel-MountList ein:

```
CD0:    Stacksize = 4096
        Priority   = 10
        GlobVec    = -1
        Unit       = 0
        Flags      = 0
        Device      = devs:pronet.device
        Surfaces    = 1
        BlocksPerTrack = 1
        Reserved    = 0
        LowCyl      = 0
        HighCyl     = 0
        FileSystem  = l:pronet-handler
        Mount       = 1
#
```

Es macht nichts, wenn Sie auf Anhieb nichts verstehen; das wichtige sind nur die beiden schräggeschriebenen Dinge:

CD0: sollten Sie durch den Namen des Devices ersetzen, das Sie erstellen wollen, in unserem Falle also 'HD4:'.

Die Nummer in der 'Unit'-Zeile gibt an, an welcher Schnittstelle die Verbindung zum anderen Rechner liegt. Geben Sie hier eine der Nummern ein, die Sie im ProNET.config-File definiert haben, z.B. '0' für den 'internal-parallel'-Treiber.

Bis jetzt weiß Ihr Amiga aber immer noch nicht, _welches_ Device denn von der externen Maschine importiert werden soll. Wir wollen 'HD0:', also schreiben Sie in die 'DEVS:ProNET.config':

HD4: HD0:

Das bedeutet, daß das ProNET-Device 'HD4:' ein Device 'HD0:' auf einem anderen Rechner anspricht, wo auch immer das sein mag.

Wichtig ist, daß beide Teile immer zusammengehören -- MountList- und ProNET.config-Eintrag. Wenn eins von beiden fehlt, können Sie das Netzwerk nicht benutzen. Ein anderer Fehler, der bei Benutzern von V1 häufig auftrat, war, daß sie das Gleiche auf dem anderen Rechner wiederholten. Das muß bzw. darf aber gar nicht sein, denn dort reicht der Server. Erst, wenn Sie dann noch Devices 'über Kreuz' rüberholen wollen, müssen Sie auch auf dem (von unserer Sicht aus) Server-Computer MountList-Einträge erstellen; bei den meisten Konfigurationen ist die zweite Maschine jedoch nur ein Datenserver.

Wiederholen Sie nun diese Schritte für jedes andere Device, das Sie importieren wollen, z.B. 'RAM:', das Sie 'RAN:' auf Ihrer Hauptmaschine nennen können, oder auch 'ZWEITERAMDISK:' -- der Name ist vollkommen egal.

Für weitere Konfigurationsmöglichkeiten sehen Sie sich das Kapitel Konfiguration an.

1.9 ProNET Installation - Utilities

pronet-run

Kopieren Sie 'pronet-run' in das 'C: '-Verzeichnis des Hauptrechners.

pronet-page

Kopieren Sie 'pronet-page' in das 'C: '-Verzeichnis des Hauptrechners.

pronet-talk

Kopieren Sie 'pronet-talk' in das 'C: '-Verzeichnis aller am Netz teilnehmenden Amigas. Dann fügen Sie überall folgende Zeile in die 'DEVS:ProNET.config' ein, wenn sie nicht bereits existiert:

pronet-talk: 1994

Die Nummer 1994 gibt die sogenannte ProNET.device Portnummer an, die völlig uninteressant ist für den Benutzer. Wichtig ist, daß sie auf allen Rechnern gleich ist.

1.10 ProNET Start des Netzwerks

Als erstes starten Sie den 'pronet-server' auf allen Server-Amigas. Dabei müssen Sie als erstes Argument die ProNET-Unitnummer übergeben, von der der Server seine Befehle erhalten soll. Sie müssten den Server also für eine andere Unit neu starten.

Dann benutzen Sie den Mount-Befehl auf den Hauptrechnern, um die Netzwerk-Devices anzumelden.

Wenn ein Hauptrechner abstürzt oder neu gestartet wird UND AUF IHM KEIN PRONET-SERVER LIEF, kann man die Devices einfach neu mounten, ohne die Server zurückzusetzen.

Die aktuelle Version vom 'internal-parallel'-Treiber arbeitet nur 100%ig korrekt, solange beide Rechner angeschaltet sind. Sobald man eine Maschine abschaltet, kann es sein, daß die andere hängt oder abstürzt!

1.11 ProNET Beispielkonfiguration

Wenn Sie den Installationsvorgang bis jetzt immer noch nicht verstanden haben, sehen Sie sich doch einfach mal meine Konfiguration an:

Mein Hauptcomputer ist ein A1200. Ein anderer Amiga, ein A500, steht unter meinem Tisch und ist mit einem A570 CD-ROM-Laufwerk verbunden. Ich benutze ProNET, um dieses CD-ROM-Laufwerk auf meinem 1200er benutzen zu können (über den Parallelport).

Zuerst die ProNET.config-Dateien:

```
----- >.8 --- >.8 --- >.8 --- >.8 --- >.8 --- >.8 --- >.8 ---
```

A1200:

pronet-device-0: internal-parallel 1 5

pronet-talk: 1994

DF2: df0:

CD0: cd0:

RAN: ram:

A500:

pronet-device-0: internal-parallel 0 5

pronet-talk: 1994

```
----- >.8 --- >.8 --- >.8 --- >.8 --- >.8 --- >.8 --- >.8 ---
```

Jetzt die MountList, die ich auf dem 1200er benutze:

```
----- >.8 --- >.8 --- >.8 --- >.8 --- >.8 --- >.8 --- >.8 ---
```

DF2: Stacksize = 4096

Priority = 10

GlobVec = -1

Unit = 0


```

Flags      = 0
Device     = devs:pronet.device
Surfaces   = 1
BlocksPerTrack = 1
Reserved   = 0
LowCyl     = 0
HighCyl    = 0
FileSystem = 1:pronet-handler
Mount      = 1
#

```

```

CD0:  Stacksize   = 4096
      Priority    = 10
      GlobVec     = -1
      Unit       = 0
      Flags      = 0
      Device     = devs:pronet.device
      Surfaces   = 1
      BlocksPerTrack = 1
      Reserved   = 0
      LowCyl     = 0
      HighCyl    = 0
      FileSystem = 1:pronet-handler
      Mount      = 1
#

```

```

RAN:  Stacksize   = 4096
      Priority    = 10
      GlobVec     = -1
      Unit       = 0
      Flags      = 0
      Device     = devs:pronet.device
      Surfaces   = 1
      BlocksPerTrack = 1
      Reserved   = 0
      LowCyl     = 0
      HighCyl    = 1
      FileSystem = 1:pronet-handler
      Mount      = 1
#

```

```

----- >.8 --- >.8 --- >.8 --- >.8 --- >.8 --- >.8 --- >.8 ---

```

Um das Netz zu starten, schalte ich beide Computer an, starte den 'pronet-server' auf dem 500er (Das Argument 0 kann ich weglassen, da es automatisch benutzt wird) und benutze 'mount cd0:' auf dem 1200er. Zusätzlich kann ich mit 'mount ran:' die externe Ram Disk mitbenutzen und mit 'mount df2:' das Diskettenlaufwerk.

1.12 ProNET pronet-talk

pronet-talk ist ein Programm, über das sich zwei Netzwerkbenutzer unterhalten können. Es öffnet ein Fenster auf der Workbench und schickt alle Eingaben sofort zu einer ProNET-Unit Ihrer Wahl. Umgekehrt werden

alle Daten, die an dieser Unit für pronet-talk ankommen, sofort in das Fenster geschrieben.

Starten Sie einfach 'pronet-talk' mit der Unitnummer als Argument. Wenn Sie es weglassen, wird '0' angenommen.

1.13 ProNET pronet-run

Mit pronet-run können Sie Shell-Befehle auf einem anderen Rechner so starten, als wenn Sie sie auf diesem direkt eingegeben hätten. Als Argument müssen Sie auch hier wieder die Unitnummer angeben, zu der der Befehl gesendet werden soll. Damit alles funktioniert, muß auf der 'Empfangsmaschine' der pronet-server laufen.

Beispiele:

```
pronet-run 0 "dir cd0:"  
pronet-run 1 "list >ram:filelist sys: all"
```

Beachten Sie, daß der Befehlsstring in Gänsefüßchen ''' gesetzt ist.

1.14 ProNET pronet-page

pronet-page schickt eine Nachricht an eine andere Maschine. Diese wird in einem kleinen Alert-Window angezeigt und muß mit der linken Maustaste bestätigt werden. Der pronet-server muß auch hier auf der Empfangsmaschine laufen.

Beispiel:

```
pronet-page 7 Hey, Unit 7... noch da??
```

Beachten Sie, daß die Nachricht nicht in Gänsefüßchen ''' steht!

1.15 ProNET Konfiguration

Es folgt der Standard-MountList-Eintrag für den 'pronet-handler':

```
XXX: Stacksize = 4096  
    Priority    = 10  
    GlobVec    = -1  
    Unit       = X  
    Flags      = 0  
    Device     = devs:pronet.device  
    Surfaces   = 1  
    BlocksPerTrack = 1  
    Reserved   = 0  
    LowCyl     = 0  
    HighCyl    = 0  
    FileSystem = 1:pronet-handler
```



```
Mount    = 1
#
```

Es gibt hier 2 Zeilen, die interessant werden könnten: Als erstes wäre da die Mount-Zeile. Sie weist den 'Mount'-Befehl an, den 'pronet-handler' sofort zu starten. Da das neue Device sofort versuchen würde, auf den Zielrechner zuzugreifen, wäre eine Verwendung in 'DEVS:DosDrivers' etwas unmöglich. Deswegen kann man mit 'Mount = 0' diesen Vorgang hinauszögern, bis das Device explizit angesprochen wird (z.B. über 'dir XXX:').

Zur HighCyl-Zeile: Es wird Ihnen vielleicht mal passieren, daß ein Network-Medium den gleichen Namen hat wie ein Medium auf dem Hauptrechner, z.B. wenn Sie zwei Amigas vernetzen, die beide eine Festplattenpartition 'Workbench:' besitzen. Wenn Sie hier eine '1' einsetzen, wird ProNET vor jeden Namen des in ein Netzwerkdevice eingelegten Mediums die jeweilige Unitnummer setzen!

1.16 ProNET Bekannte Probleme

Das A570-CD-ROM-Laufwerk hat Probleme beim Erkennen von Diskwechseln. Wenn Sie die CD wechseln, benutzen Sie einfach 'diskchange cd0:', entweder direkt auf der betreffenden Maschine oder bequemerweise übers Netz.

Große ACTION_WRITES können zu Problemen führen, weil immer ein entsprechendes Stück Speicher auf dem Server belegt werden muß. Das kann der Fall sein, wenn Sie große Files mit Hilfe der Workbench kopieren. Die meisten DOpus-ähnlichen Programme übertragen Files allerdings in kleineren Stücken, so daß es hier keine Probleme gibt.

1.17 ProNET Update von V1, Bemerkungen

Seit Version 1 hat sich eine Menge geändert. Ich möchte Sie hier auf die wichtigsten Änderungen hinweisen, damit Ihnen der Umstieg leichter fällt.

Das erste wäre das völlig neue Konzept des 'pronet.device'. Am besten löschen Sie Ihre gesamte Konfiguration und fangen noch mal ganz von vorne an. Verwechseln Sie nicht den Begriff 'Unit'. Was in V1 'Unit' genannt wurde, ist jetzt die Rechnernummer, die es auch nur beim 'internal-parallel'-Treiber gibt.

Die MountList enthält jetzt in den Feldern 'Surfaces' und 'BlocksPerTrack' eine 1. Dadurch wird gewährleistet, daß auch ReqTools-Requester die Netzwerklaufwerke anzeigen.

Beachten Sie, daß die Hilfsprogramme jetzt alle eine Unitnummer als erstes Argument benötigen. Außerdem gehört bei 'pronet-run' der Kommandostring jetzt in Gänsefüßchen.

Wenn Ihr Hauptrechner abstürzt, müssen Sie den Server nicht mehr neu starten. Es reicht, wenn Sie die Devices einfach neu mounten. Der Server wird automatisch merken, daß sie einen Reset gemacht haben.

Letzte wichtige Änderung ist die Entfernung des 'install'-Verzeichnisses. Ich fand, daß es sehr wichtig ist, wenn Benutzer zuerst die Anleitung lesen, damit sie einigermaßen verstehen, was sie benutzen.

1.18 ProNET Registration

"Ja, schon wieder ein Shareware-Programm..."

Ich bin allerdings davon überzeugt, daß Sie die Shareware-Gebühr nicht umbringen wird, denn sie ist extrem günstig ausgefallen für ein so mächtiges Programm (und überhaupt verglichen zu anderen Programmen...); sie beträgt nämlich bloß 10 DM. Ich glaube, daß ein höherer Preis für Leute, die ihren Computer nicht 24 Stunden am Tag benutzen, nicht gerechtfertigt ist; Das ist eine Tatsache, die für die meisten Leute, die ich kenne, zutrifft, mich eingeschlossen.

Wenn Sie finden, daß dieses Paket 10 deutsche Mark wert ist, schicken Sie bitte diesen Betrag oder einen entsprechenden in US\$, £, SFr oder FF (KEINE MÜNZEN bitte!) an:

Michael Krause
Mannesallee 24
D-21107 Hamburg
DEUTSCHLAND

Sie brauchen gar nicht auf ein Keyfile oder etwas ähnliches zu warten, weil Sie bereits eine voll funktionsfähige Version von ProNET V2 benutzen!

1.19 ProNET Support

Aufgrund der niedrigen Registrationsgebühr und den Kosten von 'normaler' Post ist es mir generell nicht möglich, jede Frage per Post zu beantworten. Wenn Sie eine Internet-Adresse haben, steigen Ihre Chancen auf eine Antwort in unerwartete Höhen :) Versuchen Sie bitte auch, zuerst eine Antwort in der FAQ zu finden.

Neue Versionen von ProNET werden sofort über das AmiNet, die Fred-Fish-CD-ROMs sowie verschiedene Mailboxen verbreitet. Wenn Sie eine neue Version direkt auf Diskette zugeschickt bekommen wollen, erhöht sich die Registrationsgebühr auf 15 DM.

Wenn Sie einen Bug zu berichten haben, schicken Sie einen Brief an oben genannte Adresse oder an diese Internet-Adresse:

rawstyle@online.sh.sub.de

Bitte gestalten Sie Bugreports so genau wie möglich; es ist übrigens egal, welche Maus Sie besitzen...

Übrigens scheint es, daß das Routing zu meiner EMail-Adresse sehr langsam ist, manchmal funktioniert es vielleicht gar nicht. Falls mich aber etwas erreichen sollte, wird es meistens am gleichen Tag beantwortet.

Bitte versuchen Sie außerdem, keine großen Dateien zu schicken, da ich für ein- und ausgehende Mails bezahlen muß (Falls Sie aus Deutschland mailen, brauchen Sie sich allerdings nicht zurückzuhalten). Danke!

1.20 ProNET Credits & Danksagungen

```
pronet.device,  
pronet-server,  
pronet-handler,  
pronet-talk,  
pronet-run,  
pronet-page  
und alle dazugehörigen Files von
```

Michael Krause

ParNet und das Verbindungskabel von

Matt Dillon

Beta-Tester:

```
## Christoph Dietz ##  
## Holger H. ##  
## Michael Schepers ##  
## Thomas Schwarz ##  
## Wolfgang Gutberlet ##  
## Nicola Soggia ##
```

[Vielen Dank!]

Meinen besonderen Dank an Nik Soggia <nsoggia@falcon.telnetwork.it>, der eine Lösung für das ReqTools-Problem gefunden hat.

Schließlich möchte ich noch meinen Dank allen denen aussprechen, die mir Mails geschrieben haben, die mehr als nur 'ProNET funktioniert nicht' enthielten und mich so ermutigten, an diesem Paket weiterzuarbeiten. Version 2 wäre ohne Eure Hilfe niemals entstanden!

1.21 ProNET Geschichte

Version 1 released am 01-Nov-94 (erste Version) mit:

```
pronet.device 33.9  
pronet-handler 33.5  
pronet-server 33.5  
pronet-talk 33.0  
pronet-run 33.0  
pronet-page 33.0
```

Version 2 released am 24-Jun-95 mit:

```
pronet.device 35.4
```

pronet-handler 34.2
pronet-server 34.9
pronet-talk 34.1
pronet-run 34.1
pronet-page 34.0

- Server brauchen nicht mehr neugestartet zu werden, wenn der Hauptrechner neustartet.
- Das Device freezt den Rechner nicht mehr.
- Priorität des Übertragungsprozesses kann eingestellt werden.
- Diskwechselerkennung verbessert
- komplett neue Anleitung
- ACTION_READs werden in kleinen Stücken übertragen, verbrauchen deshalb lange nicht so viel Speicher wie in V1.
- ACTION_EXAMINES konnten Speicherverlust auf dem Server erzeugen
- Komplette neues pronet.device, mit modularem Treiberkonzept
- Volumenname: "+"-Erweiterung >> Unitnummer
- C-Include-File
- alles, was ich vergessen habe, hier zu schreiben
- jede Menge Bug-Fixes, deswegen ist ProNET jetzt wesentlich stabiler

1.22 ProNET Zukunft

Ich habe vor, folgende Dinge noch in den nächsten Tagen bzw. Jahren zu erledigen:

- Treiber für Multiport-Karten
- vernetzte CON:-/RAW:-Fenster
- proserial.device für Benutzung anderer serieller Ports
- graphisches Netz-Statistikprogramm
- SANA-II-Interface
- Unmounting der ProNET-Devices
- ...und Ihre Vorschläge

1.23 ProNET Frequently Asked Questions

- Ist es möglich, das Netzwerk zu unterbrechen, um kurz etwas zu drucken und dann über eine Switchbox wieder aufs Netz zurückzuschalten?

Nein, das ist nicht möglich, und das wird es auch vermutlich nie sein, weil der Parallel-Treiber davon ausgeht, daß die Hardware-Register für den Parallelport nicht verändert werden.

- Existiert ein 'pronet.device' oder ein Treiber für zusätzliche Parallelports, z.B. die Multiface-Karte?

Nein, so etwas gibt es im Moment nicht, da ich keine Entwicklerdokumentationen hierüber finden kann. Die 'parallel.device'-Ersätze helfen nicht weiter! Vielleicht weiß jemand anders, wie das geht.

- Gibt es eine Möglichkeit, ein Programm auf einem Rechner zu starten und seine Ausgaben in einem CON:-Window auf einem anderen zu sehen?

Nein, aber das möchte ich in der nächsten Version einbauen.

- Ist es möglich, den seriellen Port eines anderen Netzrechners vom Hauptrechner aus mitzubেনutzen, um z.B. ein zweites Modem oder ein MIDI-Interface anzuschließen?

Nein, auch das wird es erst in der nächsten Version geben.
Danke an Giovanni Gigante für den Vorschlag!

- Wird der 'pronet-handler' jemals auf Kickstart 1.3 laufen?

Nein.

- Würde eine spezielle 020/030/040-Version des 'pronet.device' die Geschwindigkeit erhöhen?

Ich glaube nicht, aber bitte -- versuchen Sie sich am Sourcecode :)

- Ich habe einen Bug entdeckt: Wenn ich den Hauptcomputer resette und alle Devices neu mounte, kann ich ohne Probleme weiterarbeiten. Wenn ich allerdings den Server resette und den 'pronet-server' dann neu starte, läuft nichts mehr.

Das ist kein Bug, weil der 'pronet-server' einige wichtige Datenstrukturen für den Hauptrechner erzeugt, und diese werden gelöscht, sobald Sie den Server resetteten. Eine Folge davon ist auch, daß in 'Über-Kreuz-Netzwerken' kein Rechner einen Reset machen darf. Das ist ein essentieller Unterschied zwischen ProNET und 'richtigen' Netzwerken, wie z.B. Envoy.

- (Wo wir gerade bei Envoy sind..) Kann man das pronet.device als Treiber für SANA-II Netzwerksoftware benutzen?

Nein, weil mein Device nicht nach den SANA-II-Regeln gebaut ist. Allerdings möchte ich einen 'Aufsatz' für das pronet.device programmieren, so daß man alle ProNET-Treiber dann z.B. mit AmiTCP benutzen kann.

- Warum habe ich das Device nicht gleich SANA-II-kompatibel gemacht?

Weil ich bis vor einem Monat nicht von diesem Standard gehört habe :(

- Ich finde es sehr störend, daß die Shell, von der ich 'pronet-server' gestartet habe, danach absolut nutzlos ist.

Wie _jedes_ CLI-Programm, können Sie es mit

```
run >nil: <nil: pronet-server
starten, um alle Ausgaben zu unterdrücken und es in den Hintergrund
zu legen.
```

- Wenn ich einen Rechner abschalte, hängt der andere sofort.

Ich habe das noch nicht 'berichtigt', weil ein einzelner Rechner bei vorher aktiviertem Netz sowieso nicht mehr zu gebrauchen ist.

Hmm... bitte entschuldigen Sie, daß nicht allzuvielen Fragen dabei waren, die ich mit einem einfachen 'Ja' beantworten konnte :) Obwohl, Moment.. hier ist doch noch eine:

- Ist ProNET besser als ParNet?

<Was wohl??> :-)

1.24 ProNET Disclaimer

"Ich les' das jetzt nicht nochmal..."

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDER AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

1.25 ProNET License

Falls Sie der englischen Sprache nicht mächtig sind: Hier steht bloß, daß das Programm nur im Originalarchiv verbreitet werden darf, daß man keine Änderungen vornehmen darf, und solche Sachen.

- This license applies to the product called "ProNET", a collection of programs for the Amiga computer, published by Michael Krause under the concepts of shareware, and the accompanying documentation. The terms "Program" and "Package" below, refer to
-

this product. The licensee is addressed as "you".

- You may copy and distribute verbatim copies of the package as you receive it, in any medium, provided that you conspicuously and appropriately publish only the original, unmodified package, with all copyright notices and disclaimers of warranty intact and including all the accompanying documentation, example files and anything else that came with the original.
- Except when otherwise stated in this documentation, you may not copy and/or distribute this program without the accompanying documentation and other additional files that came with the original. You may not copy and/or distribute modified versions of this program.
- You may not copy, modify, sublicense, distribute or transfer the program except as expressly provided under this license. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the program is void, and will automatically terminate your rights to use the program under this license. However, parties who have received copies, or rights to use copies, from you under this license will not have their licenses terminated so long as such parties remain in full compliance.
- By copying, distributing and/or using the program you indicate your acceptance of this license to do so, and all its terms and conditions.
- Each time you redistribute the program, the recipient automatically receives a license from the original licensor to copy, distribute and/or use the program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.
- You agree to cease distributing the program and data involved if requested to do so by the author.

1.26 ProNET Sourcecode

Viele fragten mich, ob ich denn nicht die Sources zu ProNET rausrücken könnte. Also hier sind sie. ProNET ist komplett in Assembler geschrieben. Es gibt keine Makefiles, weil das meiste Ein-File-Programme sind. Sie sind von den gängigsten Assemblern wie PhxAss oder DevPac zu übersetzen.

Ich habe den Source vor allem beigelegt, damit Programmier-Anfänger hier vielleicht Antworten auf Fragen bezüglich AmigaDOS finden können. Ich arbeite mittlerweile mehr als ein Jahr an diesem Paket und möchte anderen damit gerne helfen.

Eine kommerzielle Verwertung der Sources ist nicht beabsichtigt. Außerdem möchte ich nicht, daß irgendjemand daraus neue ProNET-Pakete kompiliert und verbreitet, solange ich noch daran weiterarbeite. Ich werde das gestatten, wenn ich irgendwann die Amiga-Szene verlasse und aufhöre, dieses Paket weiterzuentwickeln.

Um es kurz zu fassen: Die Sources sind dabei, um etwas zu lernen.

1.27 ProNET Programmers

Ich habe diesen Teil nicht ins Deutsche übersetzt, weil Programmierer im Allgemeinen weniger Probleme mit dem Englischen haben. Deswegen spare ich mir die Arbeit :)

It is now extremely easy to add new drivers to ProNET. I realized that writing a completely new pronet.device was a bit too hard, so I introduced the modular driver concept in V2.

A ProNET driver is a simple executable which is copied into "DEVS:ProNET/". This executable will be loaded for every ProNET Unit which requires it; it doesn't have to be pure. The code is entered at the very first position, which must contain an initialization routine:

d0 contains the bytes "RST!" for identification, so that this driver can not be executed by accident.

d1 is a Flags register. It is not mapped to the Flags argument of the OpenDevice call!! Currently, this should be ZERO; and if it contains another value, your driver should return an error.

a1 contains a pointer to the configuration string behind the driver ID, for example:

```
pronet-device-7: yourdriver bla this is data for your driver
```

a1 would then contain a pointer to the null-terminated string 'bla this is data for your driver'.

a0 finally points to an array which must be filled out by your routine:

```
UBYTE ReadSignalBit
UBYTE pad0
APTR ReadQuery
APTR Read
APTR Write
```

ReadSignalBit must contain a valid signal bit for the pronet.device task. The task will include this bit into its Wait() loop and call the function pointed to by 'ReadQuery', if this bit becomes set.

Your ReadQuery routine must analyze the incoming packet and return the following values:

```
d0 length of the packet's main data
d1 ProNET destination port
d2 ProNET source port
```

If your routine comes to the conclusion that it was called by mistake and no incoming packet is pending, it must return NULL in d0! If this routine

returns something useful, it's guaranteed that the routine pointed to by 'Read' will be called after that.

The Read routine will be called with a pointer to the place to put the packet data into in a0. It must copy the incoming packet there and return. You don't have to return an error code, because this routine should always work!

Let's come to the Write routine now. It is called when the pronet.device wants your driver to transmit data to the other machine. The routine will be called with following values:

```
a0 *data1
d0.l length of data1 (guaranteed to be even)
a1 *data2
d1.l length of data2 (guaranteed to be even)
d2 destination port
d3 source port
```

Both chunks of data must be appended so that the other side thinks you've transmitted only one big chunk of data! The expression d0+d1 is always guaranteed to be lower than or equal to \$4000. If d1 contains zero, only one chunk of data is transmitted.

If your routine succeeds in transmitting the data, you must return NULL in d0; if timeout occurred or the line was busy or anything else which prevented you from transmitting, you must return a different value.

The initialization routine must preserve registers d2-d7/a2-a6 and return an error code in d0. If NULL is returned, the initialization succeeded, other values indicate an error.

Some last words: Please don't code something like a startup handshake, because one side can always reset and try to reconnect to the other one. If you can't prevent yourself from doing this, add a note to your docs that reconnect is impossible!

1.28 ProNET Unterstützte DosPackets

ProNET unterstützt die meisten neuen DosPackets, die in V36 und V39 eingeführt wurden: Vielleicht wurden Sie schon von ParNet durch so einen '2.0 Pkt ACT_PARENT_FH'-Requester genervt -- sie werden ihn nie wiedersehen! Hier ist die Liste der unterstützten DosPackets:

Name	benutzt ab OS
ACTION_FINDINPUT	
ACTION_FINDUPDATE	V33
ACTION_FINDOUTPUT	
ACTION_END	
ACTION_READ	
ACTION_WRITE	
ACTION_SEEK	
ACTION_CURRENT_VOLUME	
ACTION_SET_FILE_SIZE	V36
ACTION_LOCK_RECORD	V36

ACTION_FREE_RECORD V36

ACTION_LOCATE_OBJECT

ACTION_FREE_LOCK

ACTION_COPY_DIR

ACTION_PARENT

ACTION_SAME_LOCK V36

ACTION_CREATE_DIR

ACTION_CHANGE_MODE V36

ACTION_FH_FROM_LOCK V36

ACTION_COPY_DIR_FH V36

ACTION_PARENT_FH V36

ACTION_EXAMINE_OBJECT

ACTION_EXAMINE_NEXT

ACTION_EXAMINE_FH V36

ACTION_DELETE_OBJECT

ACTION_RENAME_OBJECT

ACTION_MAKE_LINK V36

ACTION_READ_LINK V36

ACTION_SET_COMMENT

ACTION_SET_DATE

ACTION_SET_PROTECT

ACTION_INFO

ACTION_RENAME_DISK

ACTION_INHIBIT

ACTION_FORMAT V36

ACTION_SERIALIZE_DISK V39

ACTION_MORE_CACHE

ACTION_WRITE_PROTECT

ACTION_IS_FILESYSTEM V36

ACTION_NIL

ACTION_FLUSH

ACTION_DISK_INFO

Diese Packets werden noch nicht unterstützt:

ACTION_EXAMINE_ALL V36

ACTION_EXAMINE_ALL_END V39

ACTION_SET_OWNER V39

ACTION_GET_DISK_FSSM

ACTION_FREE_DISK_FSSM

ACTION_ADD_NOTIFY V36

ACTION_REMOVE_NOTIFY V36