

IPX в Linux HOWTO

Kevin Thorpe, kevin@pricetrak.com, перевод Alex Ott ott@phtd.tpu.edu.ru v2.3, 06 May 1998

Этот документ описывает как получить, установить и настроить различные утилиты для операционной системы Linux, которые используют поддержку протокола IPX в ядре Linux.

Содержание

1 Введение	1
2 Отказ от ответственности	2
3 Относящаяся документация	2
4 Некоторые термины, используемые в этом документе	3
5 Краткое обсуждение топологии сетей IPX	5
6 Файлы в файловой системе /proc, относящиеся к IPX	6
7 IPX утилиты Greg Pages	6
8 Настройка вашей Linux-машины как маршрутизатора IPX	7
9 Настройка вашей Linux-машины как клиента NCP	9
10 Настройка вашей Linux машины как сервера NCP.	14
11 Настройка машины под Linux как клиента печати Novell	25
12 Настройка вашей Linux-машины как сервера печати Novell	25
13 Обзор команд пользователя и администрирования <i>ncrfs</i>	26
14 Настройка PPP для поддержки IPX	27
15 Тунелирование IPX через IP	29
16 Коммерческая поддержка IPX для Linux.	30
17 Некоторые часто задаваемые вопросы (FAQ)	31
18 Сообщение об авторских правах	32
19 Разное и благодарности	33

Примечание переводчика: Шлите мне любые комментарии и замечания, даже небольшие.

1 Введение

Этот IPX в Linux HOWTO. Вы должны прочитать Linux NET-3-HOWTO в добавление к этому документу.

1.1 Изменения с предыдущей версии

Изменение автора:

Большая благодарность Terry Dawson за работу над этим документом и поздравления с рождением ребенка :-)

Добавления:

Добавлено краткие сведения об IPX& Это сделано в ответ на много мешающих вопросов с списках рассылки.

Исправления/Обновления:

Новая версия ncrfs, которая сейчас поддерживает подключение к NDS. Она все еще в стадии бета тестирования и ее использование в вашей стране может быть запрещено из-за использования патентованной технологии.

Добавление поддержки администраторских прав (trustee rights) в mars_nwe. Эта возможность все еще в стадии бета тестирования.

1.2 Введение

Ядро Linux имеет совершенно новую сетевую реализацию по сравнению с другими Unix-подобными операционными системами. Возможность взглянуть по новому на разработку сетевого программного обеспечения ядра привело к тому, что ядро Linux имеет поддержку для ряда протоколов, не относящихся к tcp/ip. Протокол IPX один из тех, кто был включен в ядро.

Ядро Linux поддерживает только протокол IPX. Оно еще не поддерживает такие протоколы как IPX/RIP, SAP или NCP, эти протоколы поддерживаются другим программным обеспечением, таким как описано в этом документе.

Поддержка IPX первоначально была разработана Alan Cox <alan@lxorguk.ukuu.org.uk> и была значительно расширена Greg Page <greg@caldera.com>.

2 Отказ от ответственности

Я не знаю и не могу знать все что можно знать о сетевом программном обеспечении Linux. Пожалуйста примите к сведению и будьте осторожны, так как этот документ вероятно содержит ошибки. Пожалуйста читайте любые README-файлы, которые включены в различные пакеты программного обеспечения описанные в этом документе для более детальной и правильной информации. Я буду пытаться сохранить этот документ безошибочным и своевременным. Версии программного обеспечения даны текущие во время написания документа.

Нет способа, которым я или авторы программного обеспечения, описанного в этом документе, защитили против ваших собственных действий. Если вы настроите это программное обеспечение, даже так как описано в этом документе и это вызовет проблемы в вашей сети, тогда вы должны сами нести ответственность. Я включил это предупреждение потому-что дизайн и настройка сетей IPX не всегда является простым делом и иногда нежелательное взаимодействие с другими маршрутизаторами и файловыми серверами может быть результатом, если вы не спроектировали или настроили правильно вашу сеть. Я также включил это предупреждение, потому-что меня иногда спрашивали люди, несчастные достаточно поскольку они запоминали этот урок трудным путем.

3 Относящаяся документация

Этот документ предполагает, что вы понимаете как построить ядро Linux с правильно выбранными сетевыми опциями и что вы понимаете как пользоваться базовыми сетевыми утилитами,

такими как *ifconfig* и *route*. Если не понимаете, то вы должны прочитать документ *NET-3-HOWTO* <[NET-3-HOWTO.html](#)> в добавление к этому документу, как описаны эти утилиты.

Другие документы серии Linux HOWTO, которые могут быть полезными при работе с данным документом:

Ethernet-HOWTO <[Ethernet-HOWTO.html](#)>, который описывает детали настройки устройств Ethernet для Linux.

PPP-HOWTO <[PPP-HOWTO.html](#)>, поскольку поддержка IPX доступна в версии 2.2.0d и поздних, реализации PPP в Linux PPP.

3.1 Новые версии этого документа

Если ваша копия этого документа, устарела больше чем на два месяца, тогда я рекомендую получить более новую версию. Сетевая поддержка для Linux изменяется очень быстро с новыми расширениями и свойствами, так что и этот документ также изменяется довольно быстро. Последние версии этого документа всегда могут быть получены по анонимному ftp:

`ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/IPX-HOWTO` или: `ftp://sunsite.unc.edu/pub/Linux/`

через World Wide Web с *Сервер Linux Documentation Project* <<http://sunsite.unc.edu/LDP/linux.html>>, на странице: *IPX-HOWTO* <<http://sunsite.unc.edu/LDP/HOWTO/IPX-HOWTO.html>> или прямо от меня, <kevin@pricetrak.com>.

Он может также посылаться время от времени в группы новостей: `comp.os.linux.networking`, `comp.os.linux.answers` и `news.answers`.

3.2 Обратная связь

Пожалуйста посылайте мне любые комментарии, обновления или пожелания, <kevin@pricetrak.com>. Скорее всего я получу feedback, скорее всего я смогу обновить и скорректировать этот документ. Если вы найдете любую проблему с ним, пожалуйста напишите мне, поскольку я могу пропустить информацию посланную в группы новостей.

3.3 Поддержка списка рассылки

Существует список рассылки, учрежденный для дискуссий о различных пакетах программного обеспечения IPX для Linux, описанных в этом документе. Вы можете подписаться на них послав сообщение на адрес `listserv@sh.cvut.cz` со строкой `add linware` в теле сообщения. Чтобы послать сообщение в список рассылки, пошлите ваше сообщение на адрес `linware@sh.cvut.cz`. Я регулярно просматриваю этот список рассылки.

Архив списка рассылки находится на сервере www.kin.vslib.cz <<http://www.kin.vslib.cz/hypermail/linware/>>.

4 Некоторые термины, используемые в этом документе

Вы будете часто видеть в этом документе термины `client` and `server`. Обычно они довольно специфичны, но в этом документе я обобщил эти определения до маленьких, так что они обозначают следующее:

client

Машина или программа, которая инициирует акцию или соединение в целях использования некоторых сервисов или данных.

server

Машина или программа, которая принимает входящие соединения от множества удаленных машин и обеспечивает для них сервисы или данные.

Эти определения не очень достоверные, но они обеспечивают понимание для одноранговых систем, таких как *SLIP* или *PPP*, которые в действительности не имеют клиента и сервера.

Примечание переводчика: Я постарался перевести некоторые термины используя литературу по Novell Netware, но если я где-то ошибся прошу написать мне, адрес в начале документа.

Другие термины, которые вы увидите:

Bindery

bindery — это специализированная база данных, сохраняющая сетевую конфигурационную информацию на файловом сервере Novell. Клиенты Netware могут запрашивать *bindery* для получения информации о доступных серверах, маршрутизации и пользователях.

Frame Type (Тип фрейма)

этот термин используется для описания, что действительный протокол используется для переноса датаграмм IPX (и IP) через ваши сегменты сети ethernet. Существует четыре общих типа:

Ethernet_II

Это упрощенная версия оригинального стандарта DIX ethernet. Novell выделил формальный идентификатор протокола и это означает, что оба протокола IPX и IP могут сосуществовать в среде Ethernet_II довольно удачно. Протокол является общеиспользуемым в среде Novell и это хороший выбор.

802.3

Это I.E.E.E. протокол, определяющий механизм Carrier Sense Multiple Access with Collision Detection (CSMA/CD, Множественный доступ к среде переноса с определением столкновений). Он основан на оригинальном стандарте DIX Ethernet, с важным изменением, поле типа (идентификатор протокола) было преобразовано в поле длины. Это является причиной, почему в действительности IPX не должен работать здесь. IEEE 802.3 был спроектирован, чтобы переносить *только* фреймы IEEE 802.2, но существуют реализации которые используют этот тип для прямого переноса фреймов IPX и это замечательно работает. Избавьтесь от этого типа, только если не пытаетесь взаимодействовать с сетью, которая уже настроена для использования такого типа фреймов.

802.2

Это протокол I.E.E.E., который определяет набор процедур Logical Link Control (Контроль логических связей). Он обеспечивает упрощенный способ, позволяющий различным протоколам сосуществовать, но является довольно ограниченным в этом отношении. Novell использует неофициальную Service Address Point (точку сервисного адреса, подобную идентификатору протокола), но так как любой также использует его, это еще не представляет слишком много проблем.

SNAP

SNAP является Sub Network Access Protocol (Протокол доступа к подсетям). Этот протокол спроектирован для переноса на протоколах 802.3 и 802.2. Он расширяет многопротокольные возможности 802.2 и обеспечивает, некоторую меру совместимости с существующими типами фреймов Ethernet и Ethernet_II.

IPX

Internet Packet eXchange (Межсетевой обмен пакетами) — это протокол используемый корпорацией Novell для обеспечения межсетевой поддержки для их продукта NetWare(tm). IPX похож по функциональности на IP протокол, используемый в сообществе tcp/ip.

Сетевой адрес IPX

Это число, которое уникально идентифицирует частную сеть IPX. Обычно системой исчисления для этого числа является шестнадцатичная. Пример может выглядеть примерно так: 0x23a91002.

Внутренняя сеть IPX

Это виртуальная сеть IPX. Она является виртуальной, потому что она не соответствует физической сети. Она используется для обеспечения средства уникально идентифицирующего и адресующего особый хост IPX. В общем это полезно для хостов IPX, которые существуют больше чем в одной физической сети IPX, таких как файловые сервера. Адрес кодируется в той же самой форме, что и для физической сети IPX

RIP

Routing Information Protocol (протокол маршрутной информации) — это протокол, используемый для автоматического распространения сетевых маршрутов (routes) в сетях IPX. По функциональности сходен с RIP используемым сообществом tcp/ip.

NCP

NetWare Core Protocol (базовый протокол NetWare) — протокол сетевой файловой системы, спроектированный корпорацией Novell для своего продукта NetWare(tm). NCP похож по функциональности на NFS, используемый в сообществе tcp/ip.

SAP

Service Advertisement Protocol (Протокол объявления сервисов) — это протокол спроектированный корпорацией Novell, который используется для объявления сетевых сервисов в среде NetWare(tm).

Адрес устройства (Hardware address)

Это число, которое уникально идентифицирует хост в физической сети на уровне доступа к среде (media access layer). Примером этого являются *адреса Ethernet*. Адрес Ethernet обычно кодируется шестью шестнадцатиричными значениями, разделенными символом двоеточия, например 00:60:8C:C3:3C:0F

маршрут (route)

маршрут это путь, который ваш пакет проходит через сеть чтобы достичь своей цели.

????

5 Краткое обсуждение топологии сетей IPX

Это очень сильно упрощенное объяснение для тех людей, кто только начинает знакомится с IPX. Большие сети вероятно нарушат некоторые правила объясненные здесь. В сложных сетях IPX вы всегда должны проконсультироваться с их администраторами.

Сети IPX произвели революцию введя схему нумерованных *сетей*, в отличие от схемы IP, которая делает большее ударение на адресе *интерфейса*. Сеть — это набор оборудования, подключенного к одному и тому же сегменту локальной сети и *использующий один и тот же тип фрейма*. Различные типы фреймов в одном и том же сегменте локальной сети считаются отдельными сетями.

Каждой сети должен быть выделен адрес, который должен быть уникальным во всей локальной сети. Это обычно выполняется сервером NetWare(tm), но это может быть легко выполнено и Linux. Клиентам IPX этот адрес выдается сервером при запуске, им необходимо только знать правильный тип фрейма.

Маршрутизация между двумя сетями обычно выполняется помещением двух сетевых карт в сервер. Этот сервер затем запускает протокол RIP, который хранит таблицу маршрутизации для внутренних сетей. Периодически сервера обмениваются этими таблицами маршрутизации при широковещательных передачах. Внутри короткого промежутка времени каждый сервер 'исследует' топологию внутренней сети.

Если вам необходимо только иметь доступ к существующим серверам NetWare, то вы можете использовать команду `ipx_configure` (раздел 7.1) для автоматического определения интерфейсов

IPX, используя широковещательный запрос для того, чтобы найти сервер. Если происходит сбой, или вы хотите обеспечивать сервисы IPX, то вам будет необходимо определить интерфейсы вручную, используя программы `ipx_interface` или `mars_nwe`.

6 Файлы в файловой системе /proc, относящиеся к IPX

Существует некоторое число файлов, относящихся к поддержке IPX в Linux, расположенных в файловой системе /proc. Это:

`/proc/net/ipx_interface`

Этот файл содержит информацию об настроенных интерфейсах IPX на вашей машине. Они могут быть настроены вручную или могут быть автоматически обнаружены и настроены.

`/proc/net/ipx_route`

Этот файл содержит список маршрутов (routes), которые существуют в таблице маршрутов IPX. Эти маршруты могут быть добавлены вручную с помощью команд или добавлены автоматически демоном маршрутизации IPX.

`/proc/net/ipx`

Этот файл содержит список сокетов IPX, которые открыты для использования на машине.

7 IPX утилиты Greg Pages

Greg Page <greg@caldera.com> из Caldera Incorporated написал набор утилит настройки IPX и расширенную поддержку IPX для ядра Linux.

Расширения ядра позволяют сконфигурировать Linux как полномасштабный мост (bridge) или маршрутизатор (router) IPX. Расширенная поддержка IPX была вставлена в основную ветвь дистрибутива ядра Linux, так что вы вероятно уже имеете ее на своей системе.

Утилиты сетевой настройки позволяют вам возможность настроить ваши сетевые устройства на поддержку IPX и позволяют вам настроить маршрутизацию IPX и другие услуги под Linux. Сетевые утилиты IPX для Linux доступны с [sunsite.unc.edu <ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/ipx.tgz>](ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/ipx.tgz).

7.1 Более детально об утилитах IPX

`ipx_interface`

Эта команда используется для ручного добавления, удаления или проверки возможностей IPX на существующем сетевом устройстве. Обычно сетевым устройством является устройство Ethernet, такое как `eth0`. По крайней мере один интерфейс IPX должен быть назначен *первичным* (*primary*) интерфейсом и флаг `-p` программы делает это. Например для того, чтобы добавить устройству Ethernet `eth0` возможности IPX как первичный интерфейс IPX используя тип фрейма IEEE 802.2 сетевой адрес IPX `39ab0222` вы должны использовать:

```
# ipx_interface add -p eth0 802.2 0x39ab0222
```

Если тип фрейма отличается от типа, применяемого серверами NetWare(tm) в этой сети, то они будут прилежно игнорировать вас. Если тип фрейма правильный, но отличаются номера сети, сервера все равно будут игнорировать вас, но будут часто выражать свое недовольство на консоли сервера NetWare. Это гарантирует вам недовольство вашего администратора NetWare и может нарушить работу существующих клиентов NetWare.

Если вы получите ошибку во время выполнения этой программы и это случится из-за того, что у вас не настроен протокол tcp/ip, тогда вам необходимо вручную запустить интерфейс eth0, используя команду:

```
# ifconfig eth0 up
```

ipx_configure

Эта команда разрешает или запрещает автоматическую установку конфигурации интерфейсов и первичного интерфейса. Были замечены проблемы при использовании этой команды в сети с машинами под Windows 95.

--auto_interface

позволяет выбрать должно ли будет новое сетевое устройство автоматически сконфигурировано как устройство IPX или нет

--auto_primary

Позволяет выбрать должно ли программное обеспечение IPX автоматически выбирать первичный интерфейс или нет.

Типичный пример, для разрешения автоматической настройки интерфейсов и автоматического выбора первичного устройства используется следующая команда:

```
# ipx_configure --auto_interface=on --auto_primary=on
```

ipx_internal_net

Эта команда позволяет настраивать или убирать адрес внутренней сети. Адрес внутренней сети является опциональным, но когда он настроен, он всегда должен быть первичным интерфейсом. Для настройки сетевого адреса IPX ab000000 на узле (node) IPX 1 вы должны использовать команду:

```
# ipx_internal_net add 0xab000000 1
```

ipx_route

Эта команда позволяет вручную модифицировать таблицу маршрутизации IPX. Например, чтобы добавить маршрут на сеть IPX 39ab0222 через маршрутизатор с номером узла 00608CC33C0F в сети IPX 39ab0108 команда будет выглядеть следующим образом:

```
# ipx_route add 0x39ab0222 0x39ab0108 0x00608CC33C0F
```

8 Настройка вашей Linux-машины как маршрутизатора IPX

Если вы имеете несколько сегментов IPX, и вы желаете создать взаимодействие между ними, то вам необходимы услуги маршрутизатора. В среде Novell существуют 2 вида информации, которые необходимо распространять по сети. Это информация о сетевых маршрутах, распространяемая используя Novell RIP, и информация о сервисах, распространяемая используя Novell SAP. Любой маршрутизатор должен поддерживать оба этих протокола чтобы быть полезным в большинстве случаев.

Linux имеет поддержку обоих протоколов и может быть довольно легко настроено, чтобы выполнять функции полномасштабного маршрутизатора Novell.

Поддержка IPX в ядре Linux в действительности управляет пакетами IPX пересылаемые через сетевые интерфейсы, но она делает это согласно правилам, закодированным в таблице маршрутизации

IPX. Linux необходимы программы реализующие Novell RIP и SAP, чтобы обеспечивать правильность построения таблицы маршрутизации IPX и их периодическое обновление для отражения изменений в сетевом статусе.

Volker Lendecke <lendecke@namu01.gwdg.de> разработал демон маршрутизации *ipxripd*, который будет делать это. Описанный ниже пакет *mars_nwe* включает альтернативный демон маршрутизации.

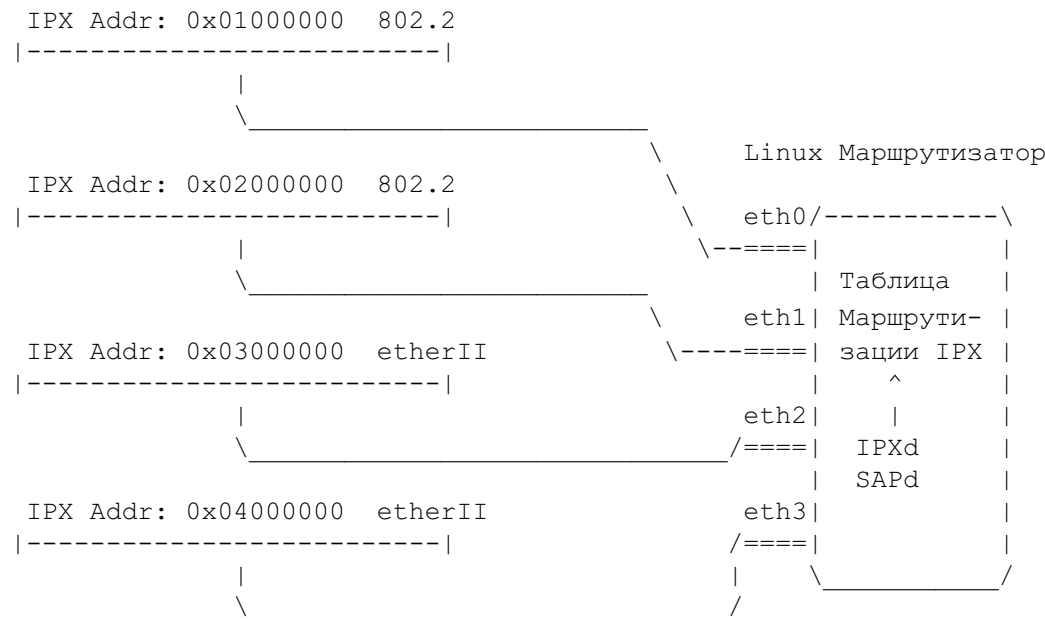
Вы можете найти *ipxripd* на: *sunsite.unc.edu* <<ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/ipxripd-0.7.tgz>>

или на сервере Volkera по адресу: *ftp.gwdg.de* <<ftp://ftp.gwdg.de/pub/linux/misc/ncpfs/ipxripd-0.7.tgz>>

Настройка возможности, чтобы ваша Linux-машина работала как маршрутизатор очень проста. Необходимо сделать следующие шаги:

1. Скомпилировать ваше ядро с поддержкой IPX, Ethernet и /proc.
2. Получить, скомпилировать и установить программу демона *ipxd*.
3. Загрузить новое ядро и убедиться, что каждая из карт Ethernet была правильно обнаружена и нет конфликтов оборудования.
4. Включить протокол IPX на каждом интерфейсе, используя команду *ipx_interface* описанную выше.
5. Запустить программу демона *ipxd*.

Рассмотрим следующую простую сеть:



Настройка для вышеуказанной сети должна выглядеть подобно этому:

```

# ipx_interface add eth0 802.2 0x0100000000
# ipx_interface add eth1 802.2 0x0200000000
# ipx_interface add eth2 etherii 0x0300000000
# ipx_interface add eth3 etherii 0x0400000000
# ipxd

```

Затем вам необходимо немного подождать и проверить ваш файл /proc/net/ipx_route и вы должны увидеть его заполненным маршрутами IPX относящихся к вашей конфигурации и другими полученными от других маршрутизаторов в сети.

8.1 Необходимо ли мне настраивать внутреннюю сеть?

Novell имеет свойство, называемое внутренняя сеть, которое используется для упрощения маршрутизации когда машина имеет больше одного сетевого устройства. Это полезно для файловых серверов, которые подключены к нескольким сетям, так как это означает, что только один маршрут необходим чтобы информационные сообщения о сервисах достигли сервера вне зависимости от того с какой сети идет попытка.

В случае конфигурации, когда вы не запустили файловый сервер и ваша машина выступает только в роли маршрутизатора IPX этот вопрос не является легким. Как было отмечено, что конфигурация для IPX/PPP работает 'лучше' если вы также настроите внутреннюю сеть.

В любом случае это легко сделать, но может потребовать перекомпиляции вашего ядра. Когда вы настраиваете его через `make config` вы должны ответить `y` когда вас спросят `Full internal IPX network` как показано:

```
...
...
Full internal IPX network (CONFIG_IPX_INTERN) [N/y/?] y
...
...
```

Для настройки интерфейса внутренней сети, используйте команду `ipx_internal_net`, описанную ранее в разделе утилит IPX. Главная предосторожность в том что надо убедиться, что сетевой адрес IPX который вы назначили является уникальным в вашей сети и что никакая другая машина или сеть не используют его.

9 Настройка вашей Linux-машины как клиента NCP

Если вы пользователь сети со смешанной технологией, которая включает протоколы IP и IPX, возможно что иногда вам захочется иметь на вашей linux-системе доступ к данным, хранимым на файловом сервере Novell в вашей сети. Novell предлагает пакет NFS сервера для своих серверов, который позволит вам организовать доступ, но если у вас маленький сервер или вы имеете маленькое количество людей, заинтересованных в доступе, может быть трудно оправдать цену коммерческого пакета.

Volker Lendecke <lendecke@namu01.gwdg.de> написал модуль ядра Linux для файловой системы, которая поддерживает подмножество Novell NCP, которая позволит вам монтировать тома Novell в файловую систему Linux без других дополнительных продуктов для вашего файлового сервера. Volker назвал пакет `ncpfs` и получил необходимую информацию в основном из книги "Netzwerkprogrammierung in C" Manfred Hill и Ralf Zessin (дополнительные детали о книге находятся в файле README в пакете `ncpfs`).

Это программное обеспечение заставляет Linux эмулировать обычную рабочую станцию Novell для файловых сервисов. Оно также включает маленькую утилиту печати, которая позволит вам печатать в очереди печати Novell (это документировано позже в разделе Клиент Печати). Пакет `ncpfs` будет работать с файловыми серверами Novell версий 3.x и поздних, он не будет работать с Novell 2.x. Клиент `ncpfs` также будет работать с Novell-совместимыми продуктами, но к сожалению некоторые продукты, объявленные как совместимые не являются достаточно совместимыми. Для использования `ncpfs` с файловыми серверами Novell 4.x, файловый сервер должен работать в режиме эмуляции `bindery`. Поддержка NDS является недвним добавлением к пакету `ncpfs` и находится в ранней стадии разработки и дополнительно к этому может быть запрещено использовать этот пакет в вашей стране, из-за использования патентованной технологии.

9.1 Где взять `ncpfs`.

Последние версии пакета `ncpfs` были спроектированы, чтобы компилироваться с ядром 1.2.13 или ядрами позже 1.3.71 (это включает 2.x.x). Если вы не используете ядро ни в одной из этих

категорий, тогда вам необходимо обновить ядро. *Kernel-HOWTO* <[Kernel-HOWTO.html](#)> в деталях описывает как сделать это.

Вы можете получить пакет *ncpfs* рпо анонимному ftp с сервера Volker по адресу: *ftp.gwdg.de* <<ftp://ftp.gwdg.de/pub/linux/misc/ncpfs/>> или *sunsite.unc.edu* <<ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs>> или других серверов-зеркал. Текущая версия на время написания была: *ncpfs-2.0.11.tgz* или *ncpfs-2.2.0.tgz*, которая добавила поддержку NDS.

9.2 Построение *ncpfs* для ядра 1.2.13.

Скомпилируйте ядро с поддержкой Ethernet и IPX

Первое, что вам необходимо сделать — убедиться что ваше ядро построено с разрешенной поддержкой IPX. В ядре версии 1.2.13 вам необходимо убедиться, что вы ответили Y на вопрос: 'The IPX protocol' как показано:

```
...
...
Assume subnets are local (CONFIG_INET_SNARL) [y]
Disable NAGLE algorithm (normally enabled) (CONFIG_TCP_NAGLE_OFF) [n]
The IPX protocol (CONFIG_IPX) [n] y
*
* SCSI support
...
...
```

Вам также необходимо убедиться, что вы включили соответствующий драйвер для вашей карты Ethernet. Если вы не знаете как сделать это, вам необходимо прочитать документ *Ethernet-HOWTO* <[Ethernet-HOWTO.html](#)>.

Вы можете продолжать компиляцию вашего ядра. Помните, что необходимо запустить программу *lilo* для инсталляции ядра, после завершения компиляции.

Разархивируйте программное обеспечение *ncpfs*

```
# cd /usr/src
# tar xvfz ncpfs-2.0.11.tgz
# cd ncpfs
```

Проверьте Makefile

Если вы намереваетесь использовать *kerneld* для автозагрузки модуля ядра *ncpfs*, тогда вы должны раскомментировать строку в *Makefile*, которая ссылается на: *KERNELD*. Если вы не уверены, что знаете что это означает вы должны прочитать *Kernel-HOWTO* <[Kernel-HOWTO.html](#)>, чтобы ознакомить себя с настройкой модулей ядра.

Скомпилируйте пакет *ncpfs*

Программное обеспечение должно быть скомпилировано просто без всякой дополнительной настройки:

```
# make
```

Скопируйте утилиты IPX куда-нибудь, если их у вас еще нет

После того как команда *make* выполнится вы должны найти все необходимые утилиты в директории *ncpfs/bin*. Вы можете использовать:

```
# make install
```

чтобы установить утилиты в выбранные Volker'ом директории. Если вы работаете на системе использующей формат ELF, то вам необходимо перезапустить `ldconfig -v` чтобы убедиться, что разделяемые библиотеки можно найти.

Скопируйте модуль *ncpfs.o* куда необходимо

Если вы компилировали для ядра 1.2.*, то вы найдете файл названный *ncpfs.o* в директории *ncpfs/bin* после выполнения *make*. Это модуль ядра *ncpfs*. Вы должны скопировать его куда необходимо. На моей системе *debian* я скопировал его в директорию */lib/modules/1.2.13/fs* и добавил *ncpfs* в файл */etc/modules*, так что он будет автоматически загружен во время загрузки системы. Если вы используете другой дистрибутив Linux, то вы должны найти место где хранятся модули ядра и скопировать скомпилированный модуль туда, или просто скопировать модуль в вашу директорию */etc*. Для загрузки модуля вручную вам необходимо выполнить команду:

```
# insmod ncpfs.o
```

9.3 Построение *ncpfs* для ядер 1.3.71+/2.0.*.

Для последних версий *ncpfs* вы должны использовать ядро 1.3.71 или более новое, это включает и ядра 2.0.*.

Если вы намереваетесь использовать ядро версии 1.3.71 или более новое, тогда исходный код *ncpfs* для ядра уже включен в стандартный дистрибутив ядра. Вы должны только ответить Y на:

```
Networking options -->
...
...
<*> The IPX protocol
...
Filesystems -->
...
...
<*> NCP filesystem support (to mount NetWare volumes)
...
...
```

Вам все равно необходимо следовать инструкциям по компиляции для ядер 1.2.*, так как вам нужно скомпилировать утилиты, но там не будет файла модуля ядра.

9.4 Настройка и использование *ncpfs*.

Настройка сетевого программного обеспечения IPX

Существует два способа настройки сетевого программного обеспечения IPX. Вы можете вручную настроить всю информацию о вашей сети IPX или можете позволить программному обеспечению определить для себя некие разумные установки с помощью команды: Существует два способа настройки:

```
# ipx_configure --auto_interface=on --auto_primary=on
```

Это должно быть разумным в большинстве случаев, но если это у вас не работает, тогда прочитайте выше раздел 'Утилиты IPX' для настройки вашего программного обеспечения вручную. Однако были замечены проблемы при использовании этой команды в сети с клиентами под Windows 95.

Проверка конфигурации

После того как ваша сеть IPX настроена вы сможете использовать команду *slist*, для того чтобы увидеть список всех файловых серверов Novell в вашей сети.

```
# slist
```

Если команда *slist* выдает сообщение подобное: `ncp_connect: Invalid argument`, тогда скорее всего ваше ядро не имеет поддержки IPX. Проверьте что вы действительно загрузили соответствующее ядро. При загрузке вы должны видеть сообщения о 'IPX' и 'ncpfs' в сообщениях системы. Если команда *slist* не показывает список серверов, тогда вам необходимо использовать метод ручной настройки сети.

Монтирование сервера или тома Novell(tm)

Если ваше сетевое программное обеспечение IPX работает нормально, то вы можете сейчас смонтировать том файлового сервера Novell в файловую систему Linux. Команда *nscrmount* используется для этих целей и требует, чтобы вы задали по крайней мере следующую информацию:

1. Имя файлового сервера
2. Идентификатор входа файлового сервера. Если для него требуется пароль, то он также необходим вам
3. Точка монтирования, т.е. куда вы хотите примонтировать том. Это должна быть существующая директория на вашей машине.
4. (по необходимости) директория на файловом сервере, которую надо смонтировать.

Существует эквивалентная команда *nscrmount* для размонтирования смонтированных файловых систем NCP. Файловые системы NCP будут отмонтироваться нормально, если ваша машине будет выключаться корректно, так что вам нет необходимости беспокоиться о размонтировании файловых систем вручную с помощью команды *nscrmount* до выполнения команды *halt* или *shutdown*.

Пример команды для монтирования файлового сервера ACCT_FS01, с идентификационным именем *guest* без пароля, в директорию */mnt/Accounts* должен выглядеть подобным образом:

```
# nscrmount -S ACCT_FS01 /mnt/Accounts -U guest -n
```

Заметим, что использование опции *-n* указывает, что для входа в систему пароль не требуется. Та же самая процедура с указанием пароля *secret* должна выглядеть так:

```
# nscrmount -S ACCT_FS01 /mnt/Accounts -U guest -P secret
```

Если вы не укажете ни опцию *-n* ни опцию *-P* у вас запросят пароль.

Проверка монтирования

Если монтирование было успешным, то вы обнаружите том доступным с идентификатором пользователя (*userid*) для которого производился вход в систему. Вы также можете пересечь структуру директорий и найти другие файлы. В качестве альтернативы вы можете использовать опцию *-V* для монтирования отдельного тома.

Так как NCP не обеспечивает значения *uid* или *gid* владельцев файлов, все файлы будут иметь разрешения и владельца присвоенных значениям точки монтирования, учтите это при разделении смонтированного тома между пользователями Linux.

Настройка автоматического выполнения монтирований

Если вы имеете нужду в постоянно смонтированных томах ncp, тогда вы захотите настроить вышеприведенные команды в ваши rc-файлы, так что все будет выполняться автоматически во время загрузки. Если ваш дистрибутив не обеспечивает некоторый способ настройки IPX, как в *debian*, тогда я рекомендую поместить эти команды в ваш файл `/etc/rc.local`, если он у вас есть. Вы можете использовать что-то подобное:

```
#
# Start the ncp filesystem
/sbin/insmod /lib/modules/1.2.13/fs/ncpfs.o

# configure the IPX network
ipx_configure --auto_interface=on --auto_primary=on

# guest login to the Accounting fileserver
ncpmount -S ACCT_FS01 /mnt/Accounts -U guest -n

#
```

Существует другой способ настройки монтирований NCP — это делается путем создания файла `$HOME/.nwclient`. Этот файл содержит детальную информацию о временных или специфических для пользователя монтирований томов NCP, которые будут выполняться регулярно. Это позволит вам хранить информацию о монтировании, так что вы можете делать их заново не указывая все параметры при каждом монтировании.

Формат этого файла довольно простой:

```
# Первая запись - запись 'сервера по умолчанию' и она используется
# когда вы не указываете сервер явно.
#
# Пользователь TERRY входит на файловый сервер DOCS_FS01 с паролем 'password'
DOCS_FS01/TERRY password
#
# Гостевой вход на файловый сервер ACCT_FS01 без пароля
ACCT_FS01/GUEST -
```

Для активизации этих монтирований вы можете использовать

```
$ ncpmount /home/terry/docs
```

для монтирования: `DOCS_FS01` с идентификатором `TERRY` в директорию `/home/terry/docs`. Заметим, что эта запись была выбрана потому что сервер не был указан в команде монтирования. Если будет использоваться следующая команда:

```
$ ncpmount -S ACCT_FS01 /home/terry/docs
```

тогда том для пользователя `GUEST` на сервере `ACCT_FS01` должен будет быть смонтирован.

Замечание: чтобы этот механизм работал, вам необходимо чтобы разрешение на файл `$HOME/.nwclient` было равно `0600`, так что вы для этого должны использовать команду:

```
$ chmod 0600 $HOME/.nwclient
```

Если пользователям не являющимися администратором разрешено использовать этот механизм, тогда команда `ncpmount` должна быть с установленным `Set Userid Root`, так что вы должны дать этой команде такое разрешение:

```
# chmod 4755 ncpmount
```

Попробуйте утилиту *nsend*

утилита для отправки сообщений пользователям Novell также включена в этот пакет, она называется *nsend* и используется как показано:

```
# nsend rod hello there
```

должна послать сообщение "hello there" пользователю вошедшему под именем "rod" на ваш "основной" файловый сервер (первая запись в вашем файле `.nwclient`). Вы можете указать другой файловый сервер, используя тот же самый синтаксис как для команды *nservermount*.

10 Настройка вашей Linux машины как сервер NCP.

Существует два пакета, которые позволяют Linux обеспечивать функции файлового сервера Novell. Они оба позволяют осуществлять доступ к файлам на вашей Linux машине для пользователей использующих клиентское программное обеспечение Novell NetWare. Пользователи могут присоединять и мапировать файловые системы как локальные диски на их машинах точно также как они могут делать на настоящем файловом сервере Novell. Вы можете хотеть попробовать оба, чтобы увидеть какой лучше отвечает вашим намерениям.

10.1 Пакет *mars_nwe*

Martin Stover <mstover@freeway.de> разработал пакет *mars_nwe* для того чтобы обеспечить в linux работу файловых сервисов и сервисов печати для клиентов NetWare.

В случае если вас удивляет имя, то оно расшифровывается как: *mars_nwe* это Martin Stovers Netware Emulator (Эмулятор Netware Мартина Стоверса).

10.1.1 Возможности *mars_nwe*.

mars_nwe реализует подмножество полного Novell NCP для файловых сервисов, основанного на дисках bindery и также сервисы печати. Также вероятно что этот пакет содержит ошибки, но много людей используют его и число ошибок постоянно уменьшается с выпуском новых версий.

10.1.2 Где взять *mars_nwe*.

Вы можете получить *mars_nwe* с [ftp.gwdg.de](ftp://ftp.gwdg.de/pub/linux/misc/ncpfs/) <<ftp://ftp.gwdg.de/pub/linux/misc/ncpfs/>> или с <<ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ncpfs/>>.

Версия в момент написания документа была: *mars_nwe-0.99.pl10.tgz*.

10.1.3 Построение пакета *mars_nwe***Построение ядра с поддержкой Ethernet и IPX**

В ядре версии 1.2.13 вам необходимо только убедиться, что вы ответили Y на вопрос: 'The IPX protocol' и N на вопрос: 'Full internal IPX network' как показано:

```
...
...
The IPX protocol (CONFIG_IPX) [n] y
...
...
Full internal IPX network (CONFIG_IPX_INTERN) [N/y/?] n
...
...
```

В более новых ядрах производится подобный процесс, но действительный текст запроса может слегка измениться.

Вам также надо убедиться, что вы включили соответствующий драйвер для вашей карты Ethernet. Если вы не знаете как сделать это, вам необходимо прочитать документ *Ethernet-HOWTO* <[Ethernet-HOWTO.html](#)>.

Затем вы можете приступить к построению вашего ядра. Помните, что надо запустить команду *lilo* для установки ядра, когда его компиляция завершится.

Разожмите пакет *mars_nwe*

```
# cd /usr/src
# tar xvfz mars_nwe-0.99.pl10.tgz
```

Построение *mars_nwe*

Процесс построения пакета очень прост. Первый шаг — просто запустить *make*, эта команда создаст для вас файл *config.h*. Затем вы должны посмотреть и отредактировать файл *config.h* если это необходимо. Это позволит настроить такие настройки как, какие директории будут использоваться и максимальное количество сессий и томов, которые будет поддерживать сервер. Самые важные выглядят так:

FILENAME_NW_INI	расположение инициализационного файла
PATHNAME_PROGS	где находятся выполнимые программы поддержки
PATHNAME_BINDERY	где будут находиться файлы 'bindery'
PATHNAME_PIDFILES	директория, куда будут записываться 'pid' файлы
MAX_CONNECTIONS	максимальное количество одновременных соединений
MAX_NW_VOLS	максимальное количество томов, которое будет поддерживать mars_n
MAX_FILE_HANDLES_CONN	максимальное количество открытых файлов на одно соединение
WITH_NAME_SPACE_CALLS	если вы хотите поддерживать клиентов ncrfs
INTERNAL_RIP_SAP	хотите ли вы поддерживать маршрутизацию rip/sap с помощью mars_n
SHADOW_PWD	хотите ли вы использовать 'теневые пароли' или нет

Значения по умолчанию должны быть нормальными, но все равно проверьте их.

Когда это сделано, выполнение команд:

```
# make
# make install
```

построит сервер и установит его в соответствующие директории. Установочный скрипт также устанавливает конфигурационный файл */etc/nwserv.conf*.

Настройка сервера

Настройка довольно проста. Вам необходимо отредактировать файл */etc/nwserv.conf*. Формат этого файла на первый взгляд может показаться загадочным, но на самом деле он довольно прост. Файл содержит некоторое количество одностроковых настроечных данных. Каждая строка разделена пробелами и начинается с числа, которое обозначает содержимое этой строки. Все символы следующие за символом '#' считаются комментарием и игнорируются. Martin обеспечивает пример настроечного файла в пакете, но представлю упрощенный пример как альтернативу для вас.

```
# ТОМА (максимум 5)
# Только том SYS является необходимым. Директория содержащая том SYS
# должна содержать директории: LOGIN, PUBLIC, SYSTEM, MAIL.
# Опция 'i' регистр букв.
```

```

# Опция 'k' преобразует все имена в запросе NCP в нижний регистр
# Опция 'm' обозначает том как сменный (полезно для cdroms и т.п.)
# Опция 'r' устанавливает том только для чтения
# Опция 'o' показывает, что том является единой файловой системой
# Опция 'P' разрешает командам использоваться как файлы
# Опция 'O' позволяет использовать пространство имен OS/2
# Опция 'N' разрешает использование пространства имен NFS
# По умолчанию в верхнем регистре.
# Синтаксис:
#   1 <Имя тома> <Путь к тому> <Опции>

1   SYS          /home/netware/SYS/          # SYS
1   DATA        /home/netware/DATA/        k      # DATA
1   CDROM        /cdrom                      kmr    # CDROM

# ИМЯ СЕРВЕРА
# Если не установлено, тогда имя машины linux hostname будет
# сконвертировано в верхний регистр и использовано. Этот пункт
# является опциональным, если не настроено будеи использовано имя
# машины.
# Синтаксис:
#   2 <Имя сервера>

2   LINUX_FS01

# АДРЕС ВНУТРЕННЕЙ СЕТИ
# Адрес внутренней сети IPX это свойство, которое упрощает
# маршрутизацию IPX для многосетевых машин (машин, которые имеют
# порты больше чем на одной сети IPX).
# Синтаксис:
#   3 <Адрес внутренней сети> [<Номер узла>]
# или:
#   3 auto
#
# Если вы используете 'auto', тогда будет использован IP адрес
# вашей машины. ЗАМЕЧАНИЕ: это может быть угрожающим, пожалуйста
# будьте уверены, что используете уникальное число.
# Адрес имеет длину 4 байта в шестнадцатиричной форме (требуется
# лидирующий 0x).

3   0x49a01010  1

# СЕТЕВОЕ УСТРОЙСТВО (A)
# Этот раздел настраивает вашу сеть IPX. Если она у вас уже
# настроена, вам этот пункт не нужен. Это тоже самое что и
# использование утилит ipx_configure/ipx_interface до запуска
# сервера.
# Синтаксис:
#   4 <Номер сети IPX> <имя устройства> <тип фрейма> [<ticks>]
#                                     Frame types: ethernet_ii, 802.2, 802.3, SNAP

4   0x39a01010  eth0  802.3  1

# СОХРАНЯТЬ МАРШРУТЫ IPX ПОСЛЕ ОКОНЧАНИЯ РАБОТЫ СЕРВЕРА
# Синтаксис:
#   5 <флаг>
#       0 = не сохранять маршруты, 1 = сохранять маршруты

```



```
5 0

# ВЕРСИЯ NETWARE
# Синтаксис:
#   6 <версия>
#   0 = 2.15, 1 = 3.11

6 1

# ОБРАБОТКА ПАРОЛЯ
# Настоящие клиенты Novell для DOS поддерживают процедуру, которая
# шифрует пароли при их изменении. Вы можете выбрать, хотите ли вы
# чтобы ваш сервер поддерживал эту процедуру или нет.
# Синтаксис:
#   7 <флаг>
#   <флаг> может быть:
#   0 force password encryption. (Клиенты не могут сменить пароль)
#   1 force password encryption, разрешить изменение нешифрованного пароля
#   7 разрешаются нешифрованные пароли, но не пустые
#   8 разрешаются нешифрованные пароли включая пустые
#   9 полностью нешифрованные пароли (не работает с OS/2)

7 1

# МИНИМАЛЬНЫЕ ПРАВА GID UID
# разрешения используемые для подсоединения без входа. Эти разрешения
# будут использоваться для файлов на присоединении к вашему
# основному серверу
# Синтаксис:
#   10 <gid>
#   11 <uid>
#   <gid> <uid> из /etc/passwd, /etc/groups

10 200
11 201

# ПАРОЛЬ АДМИНИСТРАТОРА (SUPERVISOR)
# Может быть убран после первого запуска сервера. Сервер зашифрует
# эту информацию в файл bindery после запуска. Вы должны избегать
# использования пользователя 'root' и вместо этого использовать
# другой идентификатор для администрирования файлового сервера mars
#
# Эта запись читается и шифруется в файлы bindery сервера, так что он
# необходим только при первом запуске сервера, чтобы обеспечить
# чтобы пароль не украли.
#
# Синтаксис:
#   12 <Идентификатор администратора> <имя пользователя Unix> [<пароль>]

12 SUPERVISOR terry secret

# ЗАПИСИ ПОЛЬЗОВАТЕЛЕЙ
# Этот раздел ассоциирует идентификаторы NetWare с идентификаторами
# пользователей unix. Наличие пароля является опциональным.
# Синтаксис:
#   13 <Идентификатор пользователя> <имя пользователя в Unix> [<пароль>]
```

```
13 MARTIN martin
13 TERRY terry

# НАСТРОЙКА СИСТЕМЫ "ЛЕНИВОГО" АДМИНИСТРИРОВАНИЯ
# Если у вас большое количество пользователей и вы не хотите
# беспокоиться использованием индивидуального мапирования
# пользовательских имен, как в предыдущем случае, то вы можете
# автоматически мапировать идентификаторы mars_nwe в имена
# пользователей Linux. Но в настоящее время нет способа использовать
# пароли linux, так что все пользователи настроенные таким способом
# будут пользоваться единственным паролем, указанным здесь. Я
# рекомендую не использовать это до тех пор пока вас перестанет
# беспокоить безопасность.
# Синтаксис:
#   15 <флаг> <общий пароль>
#   <флаг>:    0 - не делать автоматическое мапирование пользователей
#               1 - автоматически мапировать пользователей не указанных выше
#               99 - автоматически мапировать всех пользователей этим способом

15 0 duzzenmatta

# ПРОВЕРКА РАБОТОСПОСОБНОСТИ
# mars_nwe будет автоматически убеждаться, что определенные
# директории существуют, если установлен этот флаг
# Синтаксис:
#   16 <флаг>
#   <флаг> - 0 для нет, не делать, или 1 для да, делать проверку

16 0

# ОЧЕРЕДИ ПЕЧАТИ
# Этот раздел ассоциирует принтера NetWare с принтерами unix.
# Директории очередей должны быть созданы вручную до попытки печати.
# Директории очередей НЕ являются очередями lpd.
# Синтаксис:
#   21 <имя очереди> <директория очереди> <команда печати unix>

21 EPSON SYS:/PRINT/EPSON lpr -h
21 LASER SYS:/PRINT/LASER lpr -Plaser

# ФЛАГИ ОТЛАДКИ
# Обычно они не нужны, но могут быть полезными если вы ищите проблему.
# Синтаксис:
#   <тема отладки> <флаг отладки>
#
#   100 = IPX KERNEL
#   101 = NWSERV
#   102 = NCPSErv
#   103 = NWCONN
#   104 = start NWCLIENT
#   105 = NWBIND
#   106 = NWROUTED
#
#           0 = запрещает отладку, 1 = разрешает отладку

100 0
101 0
102 0
103 0
```

```
104 0
105 0
106 0

# ЗАПУСК NWSERV В ФОНОВОМ РЕЖИМЕ И ИСПОЛЬЗОВАНИЕ ФАЙЛА ПРОТОКОЛА
# Синтаксис:
#   200 <флаг>
#       0 = запуск NWSERV в нормальном режиме и не использовать файл протокола
#       1 = запуск NWSERV в фоновом режиме и использовать файл протокола

200 1

# ИМЯ ФАЙЛА ПРОТОКОЛА
# Синтаксис:
#   201 <файл протокола>

201 /tmp/nw.log

# ДОПОЛНЯТЬ ПРОТОКОЛ ИЛИ ПЕРЕЗАПИСЫВАТЬ
# Синтаксис:
#   202 <флаг>
#       0 = добавлять к существующему файлу протокола
#       1 = переписывать существующий файл протокола

202 1

# ВРЕМЯ ВЫКЛЮЧЕНИЯ СЕРВЕРА
# Этот раздел устанавливает время между выдачей команды SERVER DOWN и
# действительным выключением сервера.
# Синтаксис:
#   210 <время>
#       в секундах. (по умолчанию 10)

210 10

# ИНТЕРВАЛ МЕЖДУ ПЕРЕДАЧАМИ ШИРОКОВЕЩАТЕЛЬНОЙ ПЕРЕДАЧЕЙ МАРШРУТОВ
# Время в секундах между широковещательными передачами сервера.
# Синтаксис:
#   211 <время>
#       в секундах. (по умолчанию 60)

211 60

# ИНТЕРВАЛ ПРОТОКОЛИРОВАНИЯ МАРШРУТИЗАЦИИ
# Устанавливает сколько широковещательных передач произойдет до
# протоколирования маршрутизационной информации.
# Синтаксис:
#   300 <число>

300 5

# ФАЙЛ ПРОТОКОЛА МАРШРУТИЗАЦИИ
# Устанавливает имя файла протокола маршрутизации
# Синтаксис:
#   301 <имя файла>

301 /tmp/nw.routes
```

```

# ДОБАВЛЕНИЕ/ПЕРЕЗАПИСЬ МАРШРУТНОЙ ИНФОРМАЦИИ
# Устанавливает хотите ли вы добавлять информацию к существующему
# файлу протокола, или перезаписывать его.
# Синтаксис:
#   302 <флаг>
#       <flag> - 0 для дополнения, 1 для создания/перезаписи

302 1

# WATCHDOG TIMING
# Устанавливает хронометраж для наблюдательных сообщений, чтобы
# убедиться что сеть жива.
# Синтаксис:
#   310 <значение>
#       <значение> = 0 - всегда посылать наблюдательные сообщения
#                   < 0 - (-ve) для запрета наблюдений
#                   > 0 - посылать наблюдательные сообщения при
#                   падении трафика ниже 'n' ticks

310 7

# ФАЙЛ СТАНЦИЙ
# Устанавливает имя для файла станций, который определяет для каких
# машин этот файловый сервер будет выступать как первичный файловый
# сервер. Синтаксис этого файла описан в директории 'examples' исходного
# кода пакета.
# Синтаксис:
#   400 <имя файла>

400 /etc/nwserver.stations

# ОБРАБОТКА 'GET NEAREST FILESERVER'
# Устанавливает как будет обрабатываться запрос SAP 'Get Nearest
# Fileserver' (получить ближайший файловый сервер).
# Синтаксис:
#   401 <флаг>
#       <флаг>: 0 - запретить запросы 'Get Nearest Fileserver'.
#               1 - файл 'stations' перечисляет исключаемые станции.
#               2 - файл 'stations' перечисляет включаемые станции.

401 2

```

Запуск сервера

Если вы так настроили сервер, что он полагается на внешние программы для настройки вашей сети и и/или обеспечения функций маршрутизации, тогда вы должны запустить эти программы до запуска сервера. Предполагая, что вы настроили сервер, так что он настроит ваши интерфейсы и будет обеспечивать сервисы маршрутизации, тогда вам необходимо только выполнить команду:

```
# nwserver
```

Проверка работы сервера

Для тестирования сервера вы первым делом должны попытаться присоединиться и войти с клиента NetWare на вашей сети. Затем вы должны сделать CAPTURE с клиента и попытаться напечатать. Если обе этих вещи работают хорошо, тогда сервер работает.

10.2 Пакет *lwarded*

Ales Dryak <A.Dryak@sh.cvut.cz> разработал *lwarded* чтобы позволить Linux функционировать как файловый сервер на базе NCP.

Ales назвал пакет *lwarded*, сокращение для *LinWare Daemon*.

10.2.1 Возможности *lwarded*

Сервер *lwarded* обеспечивает подмножество всех функций Novell NCP. Он включает функции сообщений, но не обеспечивает возможности печати. В настоящее время он не совсем хорошо работает с клиентами Windows95 или Windows NT. Сервер *lwarded* полагается на внешние программы для выполнения функций построения и обновления таблиц маршрутизации IPX и таблиц SAP. "Плохие" клиенты могут вызвать падение сервера. Что важно, возможность трансляции имен не была включена.

Сервер работает с оболочками NETX и VLM NetWare.

10.2.2 Получение *lwarded*

Пакет *lwarded* может быть построен для любого ядра новее чем 1.2.0, я рекомендую вам использовать ядро версии 1.2.13, так как для этого не требуется заплаток (patches) для ядра. Некоторые функциональные возможности IPX изменились в ядрах версии 1.3.* и это означает, что требуются заплатки чтобы все работало правильно. Соответствующие заплатки включены в состав новых ядер, так что если вы должны использовать ядра в стадии alpha, вы все равно сможете заставить *lwarded* работать правильно.

Вы можете получить пакет *lwarded* по анонимному ftp с: [klokan.sh.cvut.cz](ftp://klokan.sh.cvut.cz/pub/linux/linware/) <ftp://klokan.sh.cvut.cz/pub/linux/linware/>

или с: [sunsite.unc.edu](ftp://sunsite.unc.edu/pub/Linux/system/network/daemons) <ftp://sunsite.unc.edu/pub/Linux/system/network/daemons> или серверов-зеркал. Текущая версия в момент написания была: `lwarded-0.95.tar.gz`.

10.2.3 Построение *lwarded*

Разожмите пакет *lwarded*

Выполните что-то подобное этому:

```
# cd /usr/src
# tar xvpfz lwarded-0.95.tar.gz
```

Построение ядра с поддержкой Ethernet и IPX

Если вы используете alpha версии ядер 1.3.*, тогда вы должны использовать ядро версии 1.3.17 или более новое, потому-что прилагаемые заплатки были сделаны для него. Ядра 1.3.* старше чем 1.3.17 требуют ручного исправления для установки (*некоторая информация о том как сделать это включена в файл INSTALL в составе пакета*). Для установки заплаток для ядра версии 1.3.17 или более новых вы должны использовать команду:

```
# make patch
```

После приложения патчей, если необходимо, следующая вещь которую вам необходимо сделать — это убедиться что ваше ядро скомпилировано с поддержкой IPX. Для ядра версии 1.2.13 вам необходимо только убедиться, что вы ответили Y на вопрос: 'The IPX protocol' как показано:

```

...
...
Assume subnets are local (CONFIG_INET_SNARL) [y]
Disable NAGLE algorithm (normally enabled) (CONFIG_TCP_NAGLE_OFF) [n]
The IPX protocol (CONFIG_IPX) [n] y
*
* SCSI support
...
...

```

В более новых ядрах процесс сходен, но текст запроса может незначительно измениться.

Вам также будет необходимо проверить, что вы включили соответствующий драйвер для вашей карты Ethernet. Если вы не знаете как сделать это, тогда вы должны прочитать документ *Ethernet-HOWTO* <[Ethernet-HOWTO.html](#)>.

Затем вы можете продолжать построение вашего ядра. Помните, что необходимо запустить *lilo* для установки ядра, после окончания процесса компиляции.

Компиляция и установка *lward*

Для компиляции *lward* вы должны сначала проверить, и если необходимо отредактировать, файл `server/config.h`. Этот файл содержит различные настройки, которые будут управлять поведением вашего сервера при запуске. Значения по умолчанию достаточно разумны, но вы можете проверить, что указанные директории для файлов протокола и файлов настройки соответствуют вашей системе.

```

# make depend
# make
# make install

```

Я заметил, что 'make depend' жалуется на то что не нашла файл `float.h` на моей системе, но все равно работает. Я также нашел, что когда я попытался компилировать с gcc 2.6.3, я изменил строку:

```
#include <net/route.h>
```

на

```
#include <net/if_route.h>
```

в `lib/ipxkern.c`, так как файл иногда меняет имя.

'make install' будет пытаться установить сервер и программы демонов маршрутизации в вашу директорию `/usr/sbin`, программу *lwpasswd* в директорию `/usr/bin`, программы работы с IPX будут установлены в директорию `/sbin` и справочные страницы будут установлены в директорию `/usr/man`. Если любое из этих директорий не подходит для вашей системы, то вы должны отредактировать файл `Makefile` и изменить конечные директории на нужные.

10.2.4 Настройка и использование *lward*

Настройка сети IPX

Первая вещь которую вы должны сделать — это настройка ваших интерфейсов Ethernet для поддержки сетей IPX, которые ваш сервер будет поддерживать. Для того чтобы сделать это

вам необходимо знать сетевые адреса IPX для каждого из сегментов LAN, какие устройства Ethernet (*eth0*, *eth1* etc.) находятся на каком сегменте, какой тип фреймов (802.3, EtherII etc.) использует каждый сегмент LAN и какой адрес внутренней сети должен использовать ваш сервер (это необходимо только если ваш сервер будет обслуживать больше одного сегмента LAN). Настройка для сервера, который находится в двух непохожих сегментах с сетевыми адресами IPX равными 23a91300 и 23a91301 и адресом внутренней сети *bdefaced* может выглядеть так:

```
# ipx_internal_net add BDEFACED 1
# ipx_interface add eth0 802.3 23a91300
# ipx_interface add eth1 etherii 23a91301
```

Запуск демонов маршрутизации

Программное обеспечение ядра в действительности делает только переключивание пакетов IPX, также как и для IP, но ядро требует дополнительных программ для управления обновлением таблицы маршрутизации. Для IPX требуются два демона и оба поставляются с *lward*: *ipxripd* управляет маршрутизационной информацией IPX, а *ipxsapd* управляет информацией SAP. Для запуска демонов вам только необходимо указать место где они могут писать свои файлы протокола:

```
# ipxripd /var/adm/ipxrip
# ipxsapd /var/adm/ipxsap
```

Настройка сервера *lward*

Существует два файла, которые вы должны настроить вручную чтобы разрешить пользователям входить на ваш сервер *lward*. Это:

/etc/lwpasswd

Это место где хранится информация о пользователях LinWare. программа *lwpasswd* используется для работы с записями. В простейшем случае файл */etc/lwpasswd* выглядит следующим образом:

```
ales:
terryd:
guest:
```

Этот формат является простым списком идентификаторов входа с последующим символом ':' и затем зашифрованная версия пароля для входа. Здесь есть несколько дополнений: отсутствие зашифрованного пароля означает вход без пароля, пользователи LinWare должны быть зарегистрированы в Linux, так что любой пользователь кроме записи в файле */etc/lwpasswd* также должен иметь запись в */etc/passwd* и только пользователь *root* является пользователем, который может менять пароль для других пользователей LinWare. Если вы вошли в систему как *root* то вы можете сменить пароль пользователя LinWare как это показано:

```
# lwpasswd rodg
Changing password for RODG
Enter new password:
Re-type new password:
Password changed.
```

/etc/lwvtab

Этот файл содержит таблицу томов LinWare и хранить информацию о том какие директории будут доступны для пользователей LinWare (этот файл похож по природе на файл NFS /etc/exports). Простой пример формата этого файла:

```
SYS          /lwfs/sys
DATA        /lwfs/data
HOME        /home
```

Формат очень прост: После имени тома через пробел следует экспортируемая директория Linux. Вы должны иметь **по крайней мере** запись для тома SYS, чтобы запустить сервер. Если вы намереваетесь для ваших DOS клиентов использовать сервер LinWare как первичный файловый сервер, тогда вы должны создать стандартную структуру директорий тома SYS в директории экспортируемой как ваш том SYS. Так как эти файлы являются патентованными и авторскими правами на них владеет корпорация Novell, вы должны иметь на них лицензию. Если ваши пользователи будут использовать файловый сервер Novell как первичный сервер, то это не является необходимым.

Запуск сервера *lward*

tada!

```
# lward
```

Это выглядит разочаровывающе, не так ли? Ок, у вас есть вопрос? Какое имя сервера будет присвоено? Если вы запустите сервер как показано выше, то имя сервера LinWare будет присвоено на основе имени возвращенного командой Linux *hostname*. Если вы хотите какое-то другое имя, тогда вы можете задать имя сервера при его старте, например команда:

```
# lward -nlinux00
```

должна запустить сервер с именем linux00.

Тестирование сервера *lward*

Самая первая вещь для проверки — это то что ваш сервер LinWare появился в выводе команды *slist* запущенной с клиента DOS на вашей сети. Программа *slist* хранится на томе SYS файлового сервера Novell, так что вы должны сделать это с какой-то машины уже подключенной куда-нибудь. Если это не произошло, то проверьте что запущены *ipxsapd* и *lward*. Если выполнение команды *slist* прошло удачно, то попытайтесь подключиться к серверу и замапировать том:

```
C:> attach linux00/ales
...
...
C:> map l:=linux00/data:
C:> l:
```

Вы должны затем нормально обращаться с новым мапированием также как с другими. Разрешения на файлы которые вы будете иметь, будут основаны на разрешениях, которые разрешены вашими правами на *linux*, которые являются параллельными вашему идентификатору LinWare.

11 Настройка машины под Linux как клиента печати Novell

Пакет *ncpfs* включает две маленькие программы, которые позволяют вам производить печать с вашей Linux-машины на принтер подключенный к серверу печати Novell. Команда *nprint* позволяет вам печатать файл на очередь печати NetWare. Команда *pqlist* позволяет вам выводить список доступных очередей печати на сервере NetWare.

Для получения и установки этих команд просто следуйте описанным ранее инструкциям относящимся к клиенту NCP.

Обе команды требуют чтобы вы указали имя пользователя и пароль, так что вы можете нормально рассматривать создание некоторых скриптов чтобы сделать задачу печати более легкой.

Пример может выглядеть так:

```
# pqlist -S ACCT_FS01 -U guest -n
# nprint -S ACCT_FS01 -q LASER -U guest -n filename.txt
```

Синтаксис схож с синтаксисом команды *ncpmount*. Вышеприведенные примеры предполагают, что файловый сервер ACCT_FS01 имеет пользователя *guest* без пароля, очередь печати называемая LASER существует и пользователю *guest* разрешено печатать в нее.

На моих машинах с Linux у меня имеется короткий скрипт на командном языке для каждого из принтеров Novell. Они могут быть использованы как фильтры печати, для того, чтобы можно было печатать используя стандартную систему печати Linux.

12 Настройка вашей Linux-машины как сервера печати Novell

Программа, которая позволяет вашей Linux-машине выступать как сервер печати в сети Netware включена в пакет *ncpfs*. Для инструкций о том как получить и построить пакет, следуйте информации выше, в разделе 'Клиент Netware'. Альтернативная поддержка включена в пакет *mars_nwe*.

12.1 Непременные условия

Настройка довольно проста, но полагается на то что вы уже выполнили настройку принтеров и они работают под Linux. Описание процесса настройки дается в *Printing-HOWTO* <[Printing-HOWTO.html](#)>.

12.2 Настройка

Когда у вас есть работающая конфигурация принтеров, и вы построили и установили утилиту *pserver*, то вам необходимо добавить команды ее запуска в ваши *rc*-файлы.

Точно какая команда будет использоваться будет зависеть от того как вы хотите управлять ею, но простейшая форма подобная следующей будет работать:

```
# pserver -S ACCT_01 -U LASER -P secret -q LASERJET
```

Этот пример говорит утилите *pserver* войти на файловый сервер ACCT_01 с именем пользователя LASER и паролем *secret* и брать задания из очереди печати LASERJET. Когда входящее задание печати будет переслано то будет использоваться команда печати по умолчанию *lpr* для переноса задания печати на демон печати Linux. Очередь печати должна быть уже определена на файловом сервере и пользователь должен иметь привелегии оператора для этой очереди.

Вы можете использовать, если хотите, любую команду Linux для принятия и печати задания печати. Аргумент *-c* позволяет вам указать точную команду печати. Например:

```
# pserver -S ACCT_01 -U LASER -P secret -q LASERJET -c "lpr -Plaserjet"
```

должен работать также как предыдущий пример за исключением того, что он будет посылать задания на конфигурацию *laserjet* из *printcap* вместо конфигурации по умолчанию.

13 Обзор команд пользователя и администрирования *ncpfs*

Недавние версии пакета *ncpfs* Volker'a, включают набор пользовательских команд и команд администрирования, которые вы сможете использовать. Эти утилиты компилируются и устанавливаются как часть инсталляционного процесса *ncpfs*, так что если у вас их еще нет, то следуйте инструкциям приведенным выше в разделе 'Клиент Novell' для построения и установки утилит.

Детальная информация приведена в поставляемых справочных страницах, но я приведу короткое описание этих команд:

13.1 Команды пользователя

ncopy

Network Copy - позволяет эффективно копировать файлы используя функцию копирования Netware вместо копирования по сети.

nprint

Network Print - позволяет вам печатать файл на очередь печати Netware на сервере Netware.

nsend

Network Send - позволяет вам послать сообщение другим пользователям на сервере Netware.

nwbols

List Bindery Objects - позволяет вам увидеть содержимое bindery на сервере Netware.

nwboprops

List Properties of a Bindery Object - позволяет вам просмотреть свойства объекта bindery Netware.

nwbpset

Set Bindery Property - позволяет вам установить свойства объекта bindery Netware.

nwbpvalues

Print Netware Bindery Objects Property Contents - позволяет вам напечатать содержимое свойства bindery Netware.

nwfsinfo

Fileserver Information - печатает некоторую общую информацию о сервере Netware.

nwpasswd

Netware Password - позволяет вам сменить пароль пользователя Netware.

nwrights

Netware Rights - показывает список прав ассоциированных с отдельным файлом или директорией.

nwuserlist

Userlist - перечисляет пользователей в настоящее время подключенных к файловому серверу Netware.

pqlist

Print Queue List - показывает содержимое очереди печати Netware.

slist

Server List - показывает список известных серверов Netware.

13.2 Утилиты администрирования

nwbocreate

Create a Bindery Object - позволяет вам создать объект bindery Netware.

nwborm

Remove Bindery Object - позволяет вам удалить объект bindery Netware.

nwbpadd

Add Bindery Property - позволяет вам установить значение существующего свойства объекта bindery Netware.

nwbprcreate

Create Bindery Property - позволяет вам создать новое свойство для существующего объекта bindery Netware.

nwbprm

Remove Bindery Property - позволяет удалить свойство из объекта bindery Netware.

nwgrant

Grant Trustee Rights - позволяет вам установить попечительские (trustee) права на директорию на файловом сервере Netware.

nwrevoke

Revoke Trustee Rights - позволяет вам удалить попечительские права с директории на файловом сервере Netware.

14 Настройка PPP для поддержки IPX

Новые версии *pppd*, демона PPP для Linux, имеют поддержку, которая позволит вам переносить пакеты IPX через последовательные соединения PPP. Вам необходим по крайней мере демон версии *ppp-2.2.0d*. Смотрите *PPP-HOWTO* <[PPP-HOWTO.html](#)> для детальной информации о том где найти его. При компиляции *pppd* вы должны убедиться что вы разрешили поддержку IPX добавив следующие две строки:

```
IPX_CHANGE = 1
USE_MS_DNS = 1
```

в файл: `/usr/src/linux/pppd-2.2.0f/pppd/Makefile.linux`.

`IPX_CHANGE` настраивает поддержку IPX в PPP. Определение `USE_MS_DNS` позволяет машинам с Microsoft Windows95 делать Name Lookups (Поиск Имен).

Настоящий фокус чтобы заставить это все работать — это знание как настраивать демона.

Существует много способов настройки, но я опишу только два о которых я получил какую-то информацию. Я никогда не пытался сделать это, так что рассматривайте это раздел как экспериментальный, и если у вас что-то получится дайте мне знать.

14.1 Настройка сервера IPX/PPP

Первая вещь которую вам нужно сделать это настроить вашу Linux-машину как сервер IP/PPP. Не паникуйте. Это не так трудно. Далее, следуйте инструкциям в *PPP-HOWTO* <[PPP-HOWTO.html](#)> и все должно быть ок. Когда вы сделаете это есть несколько простых изменений, которые вам нужны чтобы заставить работать IPX на той же конфигурации.

14.1.1 Первые шаги

Один из первых шагов которые вы должны сделать — это настройка вашей Linux машины как маршрутизатора IPX, как это описано в соответствующем разделе выше в этом документе. Вам не нужно использовать команду `ipx_route` для интерфейса `ppp` потому-что `pppd` сам настроит это для вас как он это делает с IP. Когда у вас запущен демон `ipxd` он сам будеи автоматически определять любые новые интерфейсы IPX и распространять маршруты для них. В этом случае ваша машина будет видна другим машинам автоматически при подсоединении.

14.1.2 Дизайн

Когда вы работаете как сервер, то вы несете ответственность за назначение сетевых адресов на каждое PPP-соединение, когда оно установлено. Это важный вопрос, каждое PPP-соединение будет являться сетью IPX и должно иметь уникальный сетевой адрес IPX. Это означает, что вы должны решить как вы будете назначать адреса и какими они будут. Простое решение заключается в выделении одного сетевого адреса IPX на каждое последовательное устройство, которое будет поддерживать IPX/PPP. Вы можете выдавать сетевые адреса IPX основанные на идентификаторе подключающегося пользователя, но я не вижу особенно хорошей причины чтобы делать это.

Я буду предполагать что у вас это сделано, и что вы имеете два последовательных устройства (модема), которые вы будете использовать. Адреса которые я присвоил в этом примере следующие:

устройство	Сетевой адрес IPX
-----	-----
ttyS0	0xABCDEF00
ttyS1	0xABCDEF01

14.1.3 Настройка `pppd`.

Настройте ваш файл `/etc/ppp/options.ttyS0` как показано:

```
ipx-network 0xABCDEF00
ipx-node 2:0
ipxcp-accept-remote
```

и ваш файл `/etc/ppp/options.ttyS1` как:

```
ipx-network 0xABCDEF01
ipx-node 3:0
ipxcp-accept-remote
```

Это скажет `pppd` выделить соответствующий сетевой адрес IPX для соединения когда оно установлено, установить номер локального узла в 2 или 3 и позволит удаленному узлу переписать номер удаленного узла в тот который ему нужен. Заметим что каждый из этих узлов являются шестнадцатиричными числами и требуется 0x а начале сетевого адреса, но не требуется в начале адреса узла.

Существуют другие места в которых эта информация может быть настроена. Если вы имеете только один модем, то запись должна быть расположена в файле `/etc/ppp/options`. В других случаях эта информация может быть передана через командную строку `pppd`.

14.1.4 Проверка настройки сервера

Для проверки настройки сервера вам необходим уже работающий клиент. Когда клиент подсоединяется, входит в систему и запустится `pppd`, то он присвоит сетевой адрес, проверьте номер узла клиента этого сервера и согласуйте номера узлов клиентов. Когда это будет выполнено и после того как `ipxd` определит новый интерфейс, клиент сможет устанавливать соединения IPX с удаленными машинами.

14.2 Настройка клиента IPX/PPP

В настройке клиента необходимость настройки вашей Linux-машины как маршрутизатора IPX зависит от того имеете ли вы локальную сеть для которой вы хотите выступать как маршрутизатор или нет. Если у вас автономная машина, подключенная к серверу IPX/PPP, то вам нет необходимости запускать *ipxd*, но если у вас есть локальная сеть и вы хотите чтобы все машины в этой сети могли использовать IPX/PPP маршрутизацию, то вы должны настроить и запустить *ipxd* как описано. Эта настройка более проста потому-что у вас нет множественных последовательных устройств, которые необходимо настраивать.

14.2.1 Настройка *pppd*

Простейшая конфигурация это такая, которая позволяет серверу обеспечивать всю информацию о настройке сети IPX. Эта конфигурация должна быть совместимой с конфигурацией сервера, описанной выше.

Далее вам необходимо добавить некоторые опции в ваш файл `/etc/ppp/options`:

```
ipxcp-accept-network
ipxcp-accept-remote
ipxcp-accept-local
```

Эти опции говорят *pppd* полностью работать пассивно и принимать все настроечные детали от сервера. Вы можете установить значения по умолчанию для серверов, которые не обеспечивают эти детали добавлением записей `ipx-network` и `ipx-node` сходных с конфигурацией сервера.

14.2.2 Тестирование клиента IPX/PPP

Для тестирования клиента вам необходим работающий сервер к которому можно подключиться. После того как вы подключились и запустился *pppd* вы должны увидеть детали настройки IPX на вашем устройстве `ppp0` когда вы запустите команду *ifconfig* и вы должны мочь использовать *pcrmount*.

Я не уверен должны ли вы добавлять маршруты IPX вручную, чтобы достичь удаленного файлового сервера. Вероятно да. Если кто-нибудь запустит такую конфигурацию сообщите мне я буду благодарен.

15 Тунелирование IPX через IP

Многие из вас бывают в ситуации когда у вас имеется две локальных сети Novell только с IP соединением между ними. Вы можете спросить, "Как играть в многопользовательскую игру DOOM для DOS через такое расположение?" Andreas Godzina <ag@agsc.han.de> имеет ответ на этот вопрос в виде *ipxtunnel*.

ipxtunnel обеспечивает похожее на мост (bridge) свойство для IPX позволяя пакетам IPX быть включенными в датаграммы tcp/ip, так что они могут переноситься tcp/ip соединением. Программа следит за пакетами IPX и когда пакет приходит она заталкивает его в датаграмм tcp/ip и перенаправляет его на удаленную машину, адрес IP которой вы укажете. Конечно для того чтобы это работало на машине на которую вы направляете инкапсулированные пакеты IPX также должна быть запущена копия той же версии *ipxtunnel* как и у вас.

15.1 Получение *ipxtunnel*

Вы можете получить *ipxtunnel* с [sunsite.unc.edu](ftp://sunsite.unc.edu/pub/Linux/system/network/daemons) <<ftp://sunsite.unc.edu/pub/Linux/system/network/daemons>> или серверов-зеркал.

15.2 Построение *ipxtunnel*

ipxtunnel для меня строился просто используя следующие команды:

```
# cd /usr/src
# tar xvfz ../ipxtunnel.tgz
# cd ipxtunnel
# make
```

15.3 Настройка *ipxtunnel*

Настройка *ipxtunnel* довольно проста. Допустим, что машина вашего друга называется `gau.somewhere.com` а ваша машина называется `gim.sw.edu`. *ipxtunnel* использует конфигурационный файл называемый `/etc/ipxtunnel.conf`. Этот файл позволяет указать порт UDP по умолчанию для использования в соединении `tcp/ip`, куда посылать инкапсулированные данные и на каком локальном интерфейсе должен слушать *ipxtunnel* и на который отправлять пакеты IPX. Простой конфигурационный файл должен выглядеть примерно так:

```
#
# /etc/ipxtunnel.conf для gim.sw.edu
#
# Порт UDP для использования:          (по умолчанию 7666)
port 7777
#
# Удаленная машина на которую отправлять пакеты IPX: (нет значения по умолчанию)
remote gau.somewhere.com
#
# Локальные интерфейсы на которых искать пакеты IPX: (по умолчанию eth0)
interface eth0
interface eth1
```

Очевидно, что другая машина должна иметь сходный конфигурационный файл указывая эту машину как удаленную.

15.4 Тестирование и использование *ipxtunnel*

ipxtunnel работает **подобно** мосту IPX, так что сети IPX на другом конце соединения вероятно должны быть те же самыми. Andreas никогда не тестировал *ipxtunnel* в среде, которая в действительности поддерживает файловые сервера Novell, так что если вы попробуете сделать это в среде с файловыми серверами Novell, то дайте знать Andreas работает это или нет.

Если *ipxtunnel* работает вы должны смочь запустить вашу машину с DOOM на каждом конце соединения работающего в режиме IPX и вы должны видеть друг друга.

Andreas использовал этот код только через высокоскоростные линии и он не делал никаких объявлений о производительности при связи на низких скоростях. Дайте ему знать что работает у вас а что нет.

16 Коммерческая поддержка IPX для Linux.

16.1 Caldera'a Network Desktop

Caldera Inc., производит дистрибутив Linux, который содержит диапазон коммерчески поддерживаемых расширений, включая полнофункциональную поддержку клиента Novell NetWare. Основа дистрибутива это хорошо известный дистрибутив Red Hat Linux и Caldera добавила к нему свой продукт "Network Desktop". Поддержка NetWare обеспечивает полнофункциональный клиент Novell NetWare, построенный на технологии, лицензированной у Novell Corporation. Клиент

обеспечивает полный клиентский доступ к файловым серверам Novell 3.x и 4.x и включает таких возможностей как NetWare Directory Service (NDS) и RSA кодирование.

Вы можете получить намного больше информации и информацию о приобретении с: *Web Сервер Caldera Inc* <<http://www.caldera.com/>>.

Если вы работаете внутри среды Netware 4.x и/или NDS, то сетевой клиент Caldera единственное доступное решение.

Если вы имеете критические коммерческие приложения для поддержки Novell для Linux, то продукт Caldera должен быть тем на что вы должны обратить внимание.

17 Некоторые часто задаваемые вопросы (FAQ)

Где я могу найти коммерчески поддерживаемое программное обеспечение IPX для Linux?

Caldera Corporation предлагает полностью лицензированного и полностью поддерживаемого клиента Netware 3.x и 4.x. Вы можете получить информацию о нем с *Web-сервера Caldera Inc* <<http://www.caldera.com/>>.

Работает ли программное обеспечение IPX с Arcnet/Token Ring/и т.п.?

Программное обеспечение Linux для поддержки IPX работает с интерфейсами ArcNet Token Ring. Я еще не слышал о ком-нибудь пробовавшем его с AX.25. Настройка такая же как настройка для ethernet за исключением того, что вы должны подставлять где необходимо соответствующие имена вместо 'eth0' и соответствующие адреса оборудования.

Как я могу настроить больше одного интерфейса IPX?

Если у вас больше одного сетевого интерфейса, то вы должны использовать команду *ipx_interface* для ручной настройки каждого интерфейса, вы не должны использовать 'plug n play' настройку.

Как я могу выбрать адреса IPX?

Работа сети IPX сходна, но не одинакова, с работой сети IP. Главное отличие в способе использования адресов. IPX не использует концепцию подсетей и так что типы ассоциаций которые у вас есть между сетевыми адресами и сетями являются разными. Правила довольно просты:

- Каждый сетевой адрес IPX должен быть уникальным в глобальной сети (WAN). Это включает адрес внутренней сети. Много организаций использующих IPX в глобальных сетях имеют некоторый сорт стандарта адресации, которому вы должны следовать.
- Каждая машина в индивидуальной сети должна иметь уникально присвоенный адрес. В случае сети на основе ethernet это не трудно, так как каждая карта имеет уникальный адрес. В случае IPX/PPP это означает, что вы должны убедиться, что вы выделили уникальные адреса на все машины в сети, независимо от того к какому концу соединения они подключены. Адрес машины может быть не уникальным в пределах глобальной сети, поскольку сетевой адрес используемый в комбинации с адресом машины уникально идентифицируют машину.

Что такое тип фрейма, какой тип я должен использовать?

Существуют различные типы фреймов через которые вы можете работать с IPX. Наиболее общие из них описаны в разделе 'Общие термины' этого документа (Подраздел 'Типы фреймов').

Если вы устанавливаете вашу машину в существующую сеть, тогда вы должны использовать тот тип, который уже используется, для того чтобы вы могли взаимодействовать с другими машинами в сети, но если вы устанавливаете совершенно новую сеть, то вы можете использовать любой из приведенного ряда протоколов для переноса потоков IPX. Моя рекомендация если вы настраиваете совершенно новую сеть и вам необходимо переносить потоки и IP и IPX, то использовать тип фрейма Ethernet_II.

Моя машина с Windows95 не работает с автоопределением типа фрейма?

Вероятно да, может не работать. Я мог бы здесь сделать злобный комментарий, но вместо этого я просто предполагаю что вы должны использовать ручную настройку типа фреймов вместо автоматической. Это вероятно самый лучший способ.

Почему я получаю сообщение 'invalid argument' когда я настраиваю IPX?

Вы вероятно не запустили ядро с поддержкой IPX, либо перекомпилируйте ядро, или дважды проверьте что вы действительно использовали lilo для установки и запуска нового ядра.

Почему я получаю сообщение 'package not installed' когда я настраиваю IPX?

Вы вероятно не запустили ядро, которое поддерживает IPX, либо перекомпилируйте ядро, или дважды проверьте что вы действительно использовали lilo для установки и запуска нового ядра.

Почему я получаю сообщение 'IPX support not in kernel' от *pppd*?

Вы вероятно скомпилировали IPX как модуль и не обеспечили чтобы модуль загружался до запуска *pppd*.

Как я могу экспортировать по NFS смонтированную файловую систему NCP ?

Для использования NFS для экспорта файловой системы NCP вы должны монтировать ее используя опцию `-v` команды *nscrmount*. Эта опция разрешает вам монтировать только один том файлового сервера вместо обычного монтирования всех томов. Когда вы делаете это ваш демон NFS позволит вам экспортировать эту файловую систему обычным способом.

Почему не работает *slist*, когда у меня установлена внутренняя сеть в *mars_nwe*?

Вы должны разрешить функцию 'get nearest server'. Так что запись 401 в файле `/etc/nwserv.conf` должна быть равна 0, до тех пор пока у вас нет причины не отвечать на запрос 'get nearest servers'. Если вы просто хотите заставить работать *slist* и не отвечать на каждый запрос 'get nearest server', то включите адрес внутренней сети и номер узла в файл `/etc/nwserv.stations` и установите запись в файле `/etc/nwserv.conf` в значение 2.

Работает ли пакет *ncpfs* с *mars_nwe*?

Код Martin'a и Volker'a медленно начали сближаться. Недавние версии *mars_nwe* имеют опцию разрешающую ему работать с *ncpfs*. Вы должны разрешить опцию `WITH_NAME_SPACE_CALLS` в файле `config.h` пакета *mars_nwe*.

Есть ли какое-нибудь свободное программное обеспечение для DOS для работы с *mars_nwe*?

Хитрый вопрос заслуживает хитрого ответа. Я рад что вы спросили, Martin имеет пакет который он распространяет параллельно со своим пакетом *mars_nwe*, который предлагает поддержку свободного клиента DOS для сервера *mars_nwe*. Вы можете найти его на тех же серверах что и сам сервер, и он будет называться *mars_dosutils-0.01.tgz*. Он включает исходный код на C для таких программ как *slist.exe*, *login.exe*, *map.exe* и т.п. Исходный код может быть откомпилирован с помощью Borland(tm) C.

18 Сообщение об авторских правах

IPX-HOWTO это руководство по программному обеспечению поддерживающему протокол IPX в Linux. Авторские права Terry Dawson, 1995.

Эта программа является свободной; вы можете распространять и/или модифицировать ее под действием терминов изложенных в GNU General Public License опубликованной Free Software Foundation; либо версии Лицензии, или (на ваш вкус) любой более поздней версии.

Эта программа распространяется в надежде, что она будет полезной, но БЕЗ ВСЯКОЙ ГАРАНТИИ; даже без намечаемой гарантии ТОРГОВОЙ СПОСОБНОСТИ или СОСТОЯНИЯ ЗДОРОВЬЯ ДЛЯ ОСОБЫХ ЦЕЛЕЙ. Смотрите GNU General Public License для более детальной информации. Вы должны распространять копию GNU General Public License вместе с этой программой; если нет, то пишите: Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

19 Разное и благодарности

Terry Dawson <terry@perf.no.itg.telstra.com.au> за оригинальный документ.

David E. Storey <dave@tamos.gmu.edu> и Volker Lendecke <lendecke@namu01.gwdg.de> обоим за сильно помогшим поставкой мне информации для этого документа. Gilbert Callaghan <gilbert@pokey.inviso.com>, David Higgins <dave@infra.com> и Chad Robinson <chadr@brtgate.brttech.com> каждому представившему информацию о настройке IPX/PPP. Bennie Venter <bjv@Gil-galad.paradigm-sa.com> поставившему некоторую полезную информацию относящуюся к типам фреймов. Christopher Wall <vergil@idir.net> сделавшему несколько полезных предложений для улучшения читабельности и компоновки документа. Axel Boldt <boldt@math.ucsb.edu> сделавшему несколько полезных предложений и отзывов. Erik D. Olson <eriko@wrq.com> обеспечившему несколько полезных отзывов и информацию о настройке PPP для IPX. Brian King <root@brian.library.dal.ca> представившему вопросы для раздела часто задаваемых вопросов (FAQ).

"NetWare — зарегистрированная торговая марка *Novell Corporation* <<http://www.novell.com/>>.

"Caldera — зарегистрированная торговая марка *Caldera Corporation* <<http://www.caldera.com/>>.

С уважением Kevin Thorpe.

<kevin@pricetrak.com>