

PPP HOWTO

Автор Robert Hart, hartr@interweft.com.au
Русский перевод Ilgiz Kalmetev, ilgiz@mail.rb.ru

Версия v3.0, 31 марта 1997

Этот документ рассказывает как подключить ваш Linux PC к серверу PPP, как использовать PPP для того, чтобы связать две LAN вместе и предлагает один из методов настройки вашего компьютера с Linux в качестве сервера PPP. Документ также предлагает помощь по отладке неработающих PPP соединений.

Содержание

1 Введение	2
2 IP адреса	4
3 Цели этого Документа	5
4 Версии упомянутого программного обеспечения	6
5 Другие полезные/важные документы	7
6 Краткий обзор того, что должно быть сделано для работы PPP как клиента	8
7 Конфигурирование вашего Linux Ядра	11
8 Получение нужной информации о PPP сервере	14
9 Конфигурирование вашего модема и последовательного порта	15
10 Настройка преобразования доменных имен в IP адрес (DNS)	19
11 Использование PPP и привилегии root	21
12 Настройка файлов PPP соединения	22
13 Если ваш PPP сервер использует PAP (Password Authentication Protocol)	29
14 Установка PPP соединения вручную	32
15 Автоматизация ваших соединений - Создание скриптов соединения	35
16 Тестирование вашего сценария соединения	41
17 Закрытие PPP связи	43
18 Отладка	44
19 Получение помощи, когда не знаете, что еще можно сделать	46
20 Общие проблемы, появляющиеся, когда связь заработает	47
21 Использование услуг Интернет с динамическими IP адресами	48
22 Связывание двух сетей, использующих PPP	49

23 После того, как связь установлена - скрипт /etc/ppp/ip-up	50
24 Использование /etc/ppp/ip-down	53
25 Маршрутизация в LAN	53
26 Установка PPP сервера	54
27 Использование PPP по нульмодемному (прямому последовательному) кабелю	58

Авторские права

Этот документ распространяется под GPL (GNU Public License).

Распространение

Этот документ отсылаться по мере обновления в comp.os.linux.answers Также он доступен в формате HTML на:

- *Linux Howto Index*
- *PPP-HOWTO*

Другие форматы (SGML, ASCII, postscript, DVI) доступны на <ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/other-formats>.

Так как sunsite.unc.edu обычно сильно загружен, пожалуйста, пользуйтесь соответствующими зеркалами.

Подтверждения

Огромное число людей помогло мне в подготовке этого документа. Специальная благодарность Al Longuear за непосредственное руководство по PPP (если здесь имеются ошибки, то они не оттуда), Грэг Ханкинс (ведущий системы Linux Howto) и Debi Tackett (MaximumAccess.com) за множество полезных предложений по стилю, упорядочению содержания, логичность и ясность объяснений.

В заключение, благодарю всех тех людей, которые контактировали со мной по email и предлагали комментарии. Как и для всех авторов HOWTO, удовлетворение от помощи всем - эта та плата, которую мы получаем и ее нам достаточно. Написанием этого HOWTO я возмещаю свой маленький долг, который я - и все другие пользователи Linux - должны людям, которые пишут и поддерживают выбранную нами ОС.

1 Введение

PPP (Point to Point Protocol – протокол точка-точка) - это механизм для создания и запуска IP (Internet Protocol) и других сетевых протоколов на последовательных линиях связи - будь это прямая последовательная связь (по нуль-модемному кабелю), связь поверх telnet или модемная связь по телефонным линиям (и, конечно, по линиям ISDN).

Используя PPP, вы можете подключить ваш Linux PC к PPP серверу и получить доступ к ресурсам сети, к которой подключен сервер (почти) так, как будто вы подключены непосредственно к этой сети.

Также вы можете настроить ваш Linux PC как PPP сервер, чтобы другие компьютеры могли звонить на ваш компьютер и получать доступ к ресурсам вашего локального PC и/или вашей сети.

Так как PPP - это двухточечная система, то вы можете использовать PPP соединение между двумя Linux PC для связи двух сетей (или для связи локальной сети с Internet), создавая глобальную сеть (Wide Area Network – WAN).

Самое основное различие между соединениями PPP и Ethernet состоит, конечно, в скорости - стандартное соединение Ethernet работает с максимальной теоретической скоростью 10 Мб/сек (Мб/сек - миллион бит в секунду), в то время как аналоговые модемы работают на скоростях до 56 кб/сек (кб/сек - тысяча бит в секунду).

Также, в зависимости от типа соединения PPP, могут возникать некоторые ограничения в использовании некоторых приложений и сервисов.

1.1 Клиенты и серверы

PPP строго двухточечный протокол; (технически) нет различий между машиной, которая дозвонилась, и машиной, которая приняла звонок. Однако, для наглядности, полезно рассматривать это соединение с точки зрения архитектуры "клиент-сервер".

Когда вы звоните на сайт, чтобы установить соединение PPP, вы - клиент, а машина, к которой вы подключаетесь - сервер.

Когда вы настраиваете вашу машину с Linux для приема и обработки звонков с соединением по PPP, вы устанавливаете PPP сервер.

Любой Linux PC может быть и PPP сервером и клиентом одновременно, Если у вас есть больше одного последовательного порта (и модема, если нужно). Как говорилось выше, реальных различий между клиентами и серверами в отношении соединения PPP нет.

Этот документ машину, которая иницирует звонок (которые дозваниваются), считает клиентом, а машину, которая отвечает на звонок, проверяет личность звонящего (с помощью имени пользователя, пароля и, возможно, других механизмов), считает СЕРВЕРОМ.

Использование PPP в качестве клиента, чтобы подключить одну или более машин к Интернет, вероятно, то, что интересует большинство тех людей, которые используют свои Linux PC как клиенты.

Процедура, описанная в этом документе позволит вам установить и автоматизировать ваше соединение с Интернет.

Этот документ также предоставит вам руководство по установке вашего Linux PC в качестве PPP сервера, и по связыванию двух LAN вместе (с полной маршрутизацией) с использованием PPP (это часто характеризуется как установление WAN-связи).

1.2 Различия между дистрибутивами Linux

Имеется много различных дистрибутивов Linux, и все они имеют их собственные характерные особенности и способы выполнения вещей.

В частности в Linux (и Unix) имеются два различных способа запуска компьютера, конфигурации интерфейсов и т.д.

Это стили инициализации BSD и System V. Если вы почитаете некоторые группы новостей Unix, то вы обнаружите религиозную войну между сторонниками этих двух систем. Если вам нравится такой сорт развлечений, то можете позабавиться этой игрой, тратящей ресурсы Сети, присоединяйтесь! Наиболее распространены следующие дистрибутивы:

- Slackwareкоторый использует инициализацию BSD
- Red Hat (и вышеупомянутая ассоциация Caldera)который использует инициализацию SysV (хотя в немного другой форме)
- Debianкоторый использует инициализацию SysV

Инициализация BSD обычно хранит файлы инициализации в /etc/... , это файлы:

```
/etc/rc
/etc/rc.local
/etc/rc.serial
(и возможно другие файлы)
```

С недавнего времени, некоторые схемы инициализации системы BSD используют каталог /etc/rc.d... для хранения файлов запуска, а не помещают их в /etc.

Инициализация System V хранит файлы инициализации в каталогах под /etc/... или /etc/rc.d/... и в этих подкаталогах:

drwxr-xr-x	2	root	root	1024	Jul	6	15:12	init.d
-rwxr-xr-x	1	root	root	1776	Feb	9	05:01	rc
-rwxr-xr-x	1	root	root	820	Jan	2	1996	rc.local
-rwxr-xr-x	1	root	root	2567	Jul	5	20:30	rc.sysinit
drwxr-xr-x	2	root	root	1024	Jul	6	15:12	rc0.d
drwxr-xr-x	2	root	root	1024	Jul	6	15:12	rc1.d
drwxr-xr-x	2	root	root	1024	Jul	6	15:12	rc2.d
drwxr-xr-x	2	root	root	1024	Jul	18	18:07	rc3.d
drwxr-xr-x	2	root	root	1024	May	27	1995	rc4.d
drwxr-xr-x	2	root	root	1024	Jul	6	15:12	rc5.d
drwxr-xr-x	2	root	root	1024	Jul	6	15:12	rc6.d

Если вы попытаетесь проследить, где действительно настраиваются ваш ethernet интерфейс и связанные с ним сетевые маршруты, то вам надо будет просмотреть эти файлы, чтобы найти команды, которые делают это.

1.3 Инструментальные средства конфигурации PPP, специфичные для дистрибутивов

На некоторых инсталляциях (например Red Hat и Caldera), имеется PPP система работы по телефонной линии, настроенная для X Windows. Этот HOWTO не раскрывает эти дистрибутивные специфические инструментальные средства. Если у вас есть проблемы с ними, войдите в контакт с непосредственно дистрибьюторами!

Для пользователей Red Hat 4.x, теперь имеется Red Hat PPP-TIP <http://www.interweft.com.au> в области ресурсов Linux и также на Red Hat Software <http://www.redhat.com> в области поддержки.

2 IP адреса

Каждое устройство, которое соединяется с Интернет, должно иметь собственный, уникальный IP адрес. Они назначаются централизованно соответствующей полномочной организацией каждой страны.

Если вы подключаете локальную сеть (LAN) к Internet, вы ДОЛЖНЫ использовать IP адрес из назначенного вашей сети диапазона адресов компьютеров и устройств, которые составляют вашу локальную сеть. Вы НЕ МОЖЕТЕ взять такие IP адреса, какие вам захотелось и использовать их при соединении с другими LAN (уже не говоря о Интернете). В худшем случае просто ничего не будет работать вообще и может вызывать общий бардак, когда ваш 'захваченный' IP адрес начнет конфликтовать с другим компьютером, который уже использует IP адрес, который вы выбрали с бухты-баракты.

Пожалуйста, заметьте, что IP адреса, используемые в этом документе (с некоторыми исключениями) принадлежат диапазону 'адресов неподсоединенных сетей', которые зарезервированы для использования в сетях, неподключенных к Internet.

Имеются IP адреса, которые специально выделены LAN, которые не соединяются с Интернетом. Вот эти адреса:

- Один адрес сети класса A 10.0.0.0 (сетевая маска 255.0.0.0)
- 16 адресов сетей класса B 172.16.0.0 - 172.31.0.0 (сетевая маска 255.255.0.0)
- 256 адресов сетей класса C 192.168.0.0 - 192.168.255.0 (сетевая маска 255.255.255.0)

Если для вашей локальной сети не было выделено пространство IP адресов специальной организацией вашей страны, то вам надо использовать для ваших машин один из приведенных выше сетевых адресов.

Эти адреса никогда не должны использоваться в Internet.

Однако, они могут использоваться для локальной сети ethernet на машине, которая соединяется с Интернетом. Это потому, что IP адреса фактически назначены сетевому интерфейсу, не компьютеру. Так, ваш интерфейс ethernet может использовать 10.0.0.1 (например), а когда вы подключаетесь к Интернет, используя PPP, вашему PPP интерфейсу сервером будет назначен другой (допустимый) IP адрес. Ваш PC будет связан с Интернет, а другие компьютеры вашей LAN - нет.

Однако, при использовании Linux и возможности IP Masquerade (также известный как трансляция сетевых адресов NAT) с помощью Linux и программного обеспечения ipfwadm, вы можете соединить вашу LAN с Интернет (с некоторым ограничением услуг), даже если вы не имеете допустимых IP адресов для машин вашего ethernet.

Для подробной информации как это сделать см. IP Masquerade mini-HOWTO по адресу <http://sunsite.unc.edu/mdw/HOWTO/mini/IP-Masquerade>

Для большинства пользователей, которые подключают одиночную машину к поставщику услуг Интернета через PPP, получение IP адреса (или более точно, сетевого адреса) не нужно.

Если вы желаете соединить маленькую LAN с Интернет, многие поставщики услуг Интернет (ISP) могут обеспечить вас выделенной подсетью (специфической последовательностью IP адресов) из их существующего IP адресного пространства. В качестве альтернативы, используйте IP Masquerading. Для пользователей, которые соединяют отдельный PC с Интернет через ISP, большинство поставщиков использует динамическое назначение IP адреса. То есть как часть процесса соединения, PPP сервис, с которым вы входите в контакт, сообщит вашей машине, какой IP адрес надо использовать для связи с помощью PPP интерфейса в течение текущего сеанса. Этот номер не обязательно будет таким же при последующем соединении с вашим ISP.

С динамическими IP адресами, вам не назначается один и тот же IP адрес при каждом сеансе соединения. Это имеет значение для серверных приложений на вашей Linux машине вроде sendmail, ftpd, httpd и т.д. Эти услуги основаны на той предпосылке, что компьютер, предлагающий обслуживание, доступен по тому же самому IP адресу все время (или по крайней мере тому же самому полностью квалифицированному имени области - FQDN - и что DNS доступен).

Ограничения обслуживания из-за динамического назначения IP адреса (и способы работать с ними, когда возможно) обсуждены в документе ниже.

3 Цели этого Документа

3.1 Установка PPP Клиента

Этот документ обеспечивает руководство людям, которые желают использовать Linux и PPP, чтобы дозвониться до PPP сервера и установить IP соединение, используя PPP. Предполагается, что PPP уже скомпилирован и установлен на вашей Linux машине (но кратко раскрываются реконфигурирование/перекомпиляцию вашего ядра для включения поддержки PPP).

Хотя для установления PPP соединения можно использовать `dir` (стандартный способ создания SLIP соединения), сценарии `dir` вообще-то очень сложны. По этой причине, этот документ не раскрывает использование `dir` для установки PPP соединения.

Вместо этого, этот документ описывает стандартное программное обеспечение Linux PPP (`chat/pppd`).

3.2 Связывание двух LAN или LAN с Интернет, используя PPP

Этот документ обеспечивает (основную) информацию о связывании двух LAN или LAN с Интернет, используя PPP.

3.3 Установка PPP сервера

Вы должны обратить внимание, что имеется несметное число способов установки Linux PPP сервером. Этот документ описывает один метод, используемый автором для установки нескольких маленьких PPP серверов (каждый по 16 модемов).

Этот метод, как известно, работает хорошо. Однако необязательно, что он самый лучший.

3.4 Использование PPP по прямому нульмодемному соединению

Этот документ обеспечивает краткий обзор по использованию PPP для связи двух Linux PC через нулевой(пустой) модемный кабель. Этим способом также возможно связать другие OS с Linux. Чтобы это сделать, вы должны будете проконсультироваться с документацией по операционной системе, которая вас интересует.

3.5 Этот документ в настоящее время не раскрывает ...

Компилирование программного обеспечения PPP демона. См. документацию, которая приходит с версией rpprd, который вы используете.

- Соединение и конфигурирование модема в Linux (подробно). См. Serial-HOWTO и специфическую для модема инициализацию, см. Информацию по установке модема <http://www.in.net/info/modems/index.html> для информации, которая может помочь вам сконфигурировать ваш модем.
- Использование dip для создания PPP соединений. Взамен этого используется chat
- Использование socks или IP Masquerade. Имеются прекрасные документы, уже объясняющие эти два пакета.
- Использование diald для установки автоматизированного соединения. См. документацию по diald.
- Использование EQL для создания единого канала связи по двум модемным линкам.
- Специфические для дистрибутива методы соединения по PPP (типа инструмента конфигурации сети Red Hat 4.x). См. дистрибутив для документации по используемым методам.
- Возрастающее число инструментальных средств, доступных для автоматизации установления PPP. См. соответствующую документацию.

4 Версии упомянутого программного обеспечения

В этом HOWTO предполагается, что вы используете ядро Linux 1.2.x с PPP 2.1.2 или Linux 1.3.X/2.0.x и PPP 2.2.

На время написания последняя официальная версия PPP доступного для Linux - rpp-2.2f. Новая версия (rpp-2.3) - все еще в стадии бета-тестирования.

Возможно использовать PPP 2.2.0 с ядром 1.2.13. Это требует патчей для ядер. Рекомендуется для пользователей с ядром версии 1.2.13 поставить на rpp-2.2, поскольку он включает исправления нескольких ошибок и расширения.

Также вы должны обратить особое внимание на то, что вы не можете использовать PPP 2.1.2 с ядром Linux версий 2.0.X.

Пожалуйста, обратите внимание, что этот документ не покрывает проблемы, являющиеся результатом использования загружаемых модулей для ядра Linux 2.0.x., пожалуйста, см. kernel mini-HOWTO и документацию по ядру/модулю 2.0.x (в Linux 2.0.x исходное каталог /usr/src/linux/Documentation/...).

Поскольку этот документ разработан для помощи новым пользователям, вам строго рекомендуется использовать соответствующие версию ядра Linux и PPP, про которые известно, что вместе они работают устойчиво.

5 Другие полезные/важные документы

Советую почитать пользователям:

- документация, которая приходит с PPP пакетом;
- man pppd и man chat;
- Linux Network Administration Guide (NAG); см. *The Network Administrators Guide*
- Net-2/3 HOWTO; см. *Linux NET-2/3-HOWTO*
- документация по ядру Linux, установленная в /usr/src/linux/Documentation, во время установки исходных текстов Linux;
- страница информации по установке модема - см. *Информацию по установке модема*
- Превосходные книги Unix/Linux, изданные O'Reilly и Ассоциацией. См. *O'Reilly and Associates On-Line Catalogue*. Если вы плохо знакомы с Unix/Linux, сбегайте (не сходите) к вашему самому близкому компьютерному книжному магазину и купите их немедленно!
- PPP-FAQ, поддерживаемый Al Longuear, доступный из *Linux PPP-FAQ*. Он содержит много полезной информации в формате вопрос/ответ, что является очень полезным при выяснении, почему PPP не работает (правильно).
- возрастающее количество книг по Linux от различных издательств и авторов; вы стесняйтесь проверять ценность этих книг.
- Разработка Linux и дистрибутивов имеют стремительную тенденцию развития, в то время, как выпуск книг в новой редакции (вообще) намного медленнее! Покупка превосходной книги (и таких много), которая сегодня является устаревшей, вызовет у новых пользователей значительное беспокойство и расстройство.

Самая лучшая общая отправная точка для Linux документации - домашняя страница *Linux Documentation Project*. HOWTO имеет тенденцию пересматриваться достаточно регулярно.

Хотя вы можете использовать этот документ для создания вашего соединения PPP без необходимости читать любой из этих документов, в таком случае вы не будете понимать, что именно вы делаете! Также вы сможете самостоятельно сформулировать проблемы (или по крайней мере задавать более интеллектуальные вопросы в группах новостей comp.os.linux ... или списках почтовой рассылки по Linux).

Эти документы (также как и различные другие, включая RFCs) обеспечивают дополнительные и более детальные объяснения, чем можно сделать в этом HOWTO.

Если вы соединяете LAN с Интернет, используя PPP, вы должны обладать приемлемым уровнем знаний о работе с TCP/IP сетями. В дополнение к документам выше вы найдете много полезного в книгах O'Reilly "Сетевое администрирование TCP/IP" и "Построение файберволов Интернет"!

5.1 Полезные списки почтовой рассылки Linux

Имеются множество списков почтовой рассылки Linux, которые функционируют как средство связи между пользователями многих уровней знания. Во что бы то ни стало подпишитесь на те, что интересуют вас, и contribute your expertise and views.

Мудрость: некоторые списки специально нацелены на "гуру" и/или специфические темы. Хотя никто не пожалуется, если вы подписались 'lurk' (но не отправляете по почте сообщения), но

вам наверняка придется выслушать резкости (если не флейм), если вы отправляете 'чайниковские' вопросы в несоответствующие списки рассылки.

Это - не потому, что гуру ненавидят новых пользователей, а потому, что эти списки должны обсуждать специальные проблемы специальных уровней сложности.

Во что бы то ни стало подпишитесь на списки, которые предлагают открытую подписку, но ваши комментарии должны соответствовать теме списка!

Хорошая отправная точка для почтовых списков рассылки Linux - *Linux Mailing List Directory*

6 Краткий обзор того, что должно быть сделано для работы PPP как клиента

Этот документ содержит много информации - и с каждой версией ее объем возрастает!

Как следствие, этот раздел стремится обеспечить краткий обзор действий, которые вы должны будете выполнить, чтобы подключить вашу Linux систему к PPP серверу в качестве клиента.

6.1 Получение/установка программного обеспечения

Если ваш дистрибутив Linux не включает программное обеспечение PPP, то возьмите его из Linux *PPP демона*.

Это последняя официальная версия на время написания. Однако, вы возьмите в этом месте последнюю доступную версию (ppp-2.3 находится в стадии бета на время написания и должен быть скоро выпущен).

PPP пакет содержит инструкции по компиляции и установке программного обеспечения, так что это не объясняется в HOWTO!

6.2 Вкомпилированная поддержка PPP в ядре

Linux оперирует PPP двумя частями

- PPP демон упомянутый выше
- ядерная поддержка для PPP

Многие дистрибутив, кажется, по умолчанию обеспечивают встроенный в ядро PPP в своих ядрах, но другие этого не делают.

Если при начальной загрузке в сообщениях вашего ядра есть что-нибудь вроде

```
PPP Dynamic channel allocation code copyright 1995 Caldera, Inc.  
PPP line discipline registered.
```

то ваше ядро имеет встроенную поддержку PPP.

Это говорит о том, что вы вероятно захотите скомпилировать ваше собственное ядро вместо пришедшего с дистрибутивом для обеспечить наиболее эффективное использование ваших аппаратных ресурсов системой. Стоит вспомнить, что ядро не может быть выгружено из памяти, и потому уменьшение ядра создает преимущества для машин с ограниченным размером памяти.

Этот документ обеспечивает минимальные инструкции по перестройке ядра в разделе "Конфигурированием вашего ядра Linux".

Для подробностей, см. Kernel-HOWTO на *Linux Kernel HOWTO*.

6.3 Получение информации от вашего ISP

Имеется почти бесконечное число способов, которыми PPP сервер может быть установлен. Чтобы соединиться с вашим ISP (или корпоративным PPP сервером, чтобы обратиться к вашему intranet), вы должны будете получить информацию о том, как функционирует PPP сервер.

Поскольку вы используете Linux, вы можете испытать затруднения с получением исчерпывающей справки от ISP (и работы сайта, основанного на intranet PPP сервере), которые в большинстве ориентируются на клиентов с MS WINDOWS.

Однако, быстро возрастает число ISP, использующих в своей работе Linux, - и Linux также проникает в корпоративную среду, так что вам может повезти в решении ваших проблем.

Раздел "Получение нужной информации о PPP сервере" рассказывает, что вы должны узнать о PPP сервере, к которому вы собираетесь подсоединиться - и как выяснить эту информацию.

6.4 Конфигурирование вашего модема и последовательного порта

Для того чтобы соединиться с PPP сервером и получать наилучшую возможную скорость передачи данных, ваш модем должен быть правильно сконфигурирован.

Точно так же последовательные порты на вашем модеме и компьютере должны быть установлены правильно.

Как это сделать рассказывается в разделе "Конфигурирование вашего модема и последовательного порта".

6.5 Настройка преобразования доменных имен в IP адрес (DNS)

В дополнение к файлам, которые запускают PPP и выполняют автоматизированный вход на PPP сервер, есть ряд текстовых конфигурационных файлов, которые должны быть установлены на вашем компьютере для того, чтобы преобразовать имена типа www.interweft.com.au в IP адрес, который используется фактически для установления контакта с этим компьютером. Это файлы:

```
/etc/resolv.conf  
/etc/host.conf
```

Раздел "Настройка преобразования доменных имен в IP адрес" содержит подробности по их настройке.

В частности вы не должны запускать сервер имен на вашем Linux PC, чтобы соединиться с Интернетом (хотя можете захотеть). Все, что вам нужно знать, это IP адрес как минимум одного сервера имен, который вы можете использовать (предпочтителен сервер имен одного из ваших ISP).

6.6 PPP и привилегии root

Так как установление связи PPP между вашим компьютером Linux и другим PPP сервером требует манипуляций с сетевыми устройствами (интерфейс PPP - это сетевой интерфейс) и таблицей маршрутизации ядра, то pppd требует привилегий root.

Для подробностей см. раздел "Использование PPP и привилегии root".

6.7 Проверка PPP файлов вашего дистрибутива и установка опций PPP

Имеется ряд конфигурационных и файлов звонилки, которые должны быть установлены для работы с PPP. В дистрибутиве PPP имеются примеры, и в этом разделе показано, какие файлы вы должны иметь:

```
/etc/ppp/options  
/etc/ppp/scripts/ppp-on
```

```
/etc/ppp/scripts/ppp-on-dialer  
/etc/ppp/options.tpl
```

Вам возможно понадобится создать некоторые дополнительные файлы в зависимости от того, что вам нужно от PPP:

```
/etc/ppp/options.ttyXX  
/etc/ppp/ip-up  
/etc/ppp/pap-secrets  
/etc/ppp/chap-secrets
```

Кроме того, демон PPP может использовать большое количество опций командной строки, и важно использовать правильные; так что этот раздел проведет вас по стандартным опциям PPP и поможет выбрать опции, которые вы должны использовать.

Для подробностей см. "Настройка файлов PPP соединения".

6.8 Если ваш PPP сервер использует PAP (Password Autentificaion Protocol)

Многие ISP и корпоративные PPP серверы используют PAP. Если ваш сервер не требует для вы использовали PAP (если вы можете зарегистрировать вручную и получать стандартную текстовую подсказку входа в систему (логин/пароль), значит он не использует PAP), то вы можете спокойно проигнорировать этот раздел.

Вместо того для зарегистрироваться на таком сервере, используя имя пользователя и пароля когда их ввод запрошен сервером, PPP сервер, использующий PAP не требует обычного ввода логина/пароля для входа в систему.

Вместо этого информация установления подлинности пользователя идет как часть протокола управления связи (LCP), который является первым шагом установления связи PPP.

Раздел "Если ваш PPP сервер использует PAP (Password Autentificaion Protocol)" предоставляет информацию о файлах, которые вы должны установить, чтобы установить связь PPP, используя PAP.

6.9 Соединение с PPP сервером вручную

При установке основных файлов хорошая идея проверить их, соединившись (используя `minicom` или `seyon`) и запустив `pppd` на вашем Linux PC вручную.

См. раздел "Установка PPP соединения вручную" для деталей по этому вопросу.

6.10 Автоматизация вашего PPP соединения

Как только вы сможете войти в систему вручную, вы можете переделать установки скриптов так для автоматизировать процесс соединения.

Раздел "Автоматизация ваших соединений - Создание скриптов соединения" разъясняет установку необходимых скриптов, акцентируя внимание на `chat` и создании сценариев входа в систему через PPP сервер.

Этот раздел обсуждает скрипты определения имени/пароля пользователя, а также написание скриптов для аутентификации PAP/CHAP на сервере.

6.11 Завершение связи

Нужно уметь закрывать работающее соединение.

Это описывается в разделе "Завершение PPP связи".

6.12 Если у вас есть проблемы

Многие люди затрудняются установить PPP связь. Разнообразием PPP серверов и их требований к установке соединения очень много. Точно так же имеется много опций PPP - и некоторые их комбинации не работают вместе.

К проблемам регистрации и запуска PPP обслуживания также добавляются проблемы с модемами и фактическими телефонными линиями!

Раздел "Решение проблем" предлагает некоторую основную информацию об общих ошибках, как их изолировать и исправить.

Он НЕ ставит своей задачей дать больше, чем только основы. Al Longuear поддерживает PPP-FAQ, который содержит много подробной информации по этой теме!

6.13 После того, как связь установлена

Как только связь PPP заработает (особенно, как только IP уровень станет работоспособным), Linux PPP может запускать автоматически (как пользователь root) скрипт, чтобы выполнить любую функцию, которую вы можете записать для выполнения в скрипт.

В разделе "После того, как связь установлена" изложена информация о скрипте /etc/ppp/ip-up, параметрах, которые он получает от PPP, и как его использовать, чтобы делать вещи типа скачивания вашего email от вашего ISP, посылки поставленной в очередь почты и все такое прочее.

6.14 Проблемы со стандартными IP услугами для PPP связи с динамическим IP адресом

Как отмечено во введении, динамические IP адреса влияют на возможность работы вашего Linux PC в качестве сервера в Интернет.

Раздел "Проблемы со стандартными IP услугами для PPP связи с динамическим IP адресом" обеспечивает информацию о таком влиянии на (основные) услуги и что вы можете сделать (если что-нибудь можно сделать), чтобы преодолеть это влияние.

7 Конфигурирование вашего Linux Ядра

Для того чтобы использовать PPP, ваше ядро Linux должно быть скомпилировано с включенной поддержкой PPP. Возьмите исходный текст для вашего ядра Linux, если у вас его еще нет - он лежит в /usr/src/linux в стандартной файловой системе Линукса.

Проверьте этот каталог - многие дистрибутивы Linux, устанавливая дерево исходных текстов (файлы и подкаталоги) в процессе инсталляции.

При загрузке, ваше ядро Linux печатает много полезной информации.

В ней можно найти информацию о поддержке PPP, если ядро включает ее. Чтобы просмотреть эту информацию, посмотрите ваш файл syslog или используйте `dmesg | less`, чтобы отобразить информацию на экране. Если ваше ядро включает поддержку PPP, вы увидите строки типа

PPP Dynamic channel allocation code copyright 1995 Caldera, Inc.
PPP line discipline registered.

(это для версий ядра Linux 2.0.x).

Исходные тексты ядра Linux могут быть получены по ftp с sunsite.unc.edu или зеркал.

7.1 Установка исходных текстов ядра Linux

Ниже перечислены краткие инструкции для получения и установки исходных текстов ядра Linux. Полная информация может быть получена *Linux Kernel-HOWTO*.

Для того чтобы установить и скомпилировать ядро Linux, вы должны зарегистрироваться как root.

1. Перейдите в каталог /usr/src : `cd /usr/src`
2. Проверьте наличие каталога с исходными текстами /usr/src/linux
3. Если у вас нет исходных текстов, то скачайте их с *Интернет сайта* или ближайшего зеркала. Если Вы ищете более ранние версии ядра (типа 1.2.X), то они хранятся в каталоге *старого ядра Linux*.
4. Выберите соответствующее ядро - обычно самое современное доступно для всех. Отыщите его и поместите исходный tar файл в /usr/src.

Обратите внимание: файл 'tar' - архив - возможно сжатый (так как исходные тексты ядра Linux это файлы tar), содержащий множество файлов в ряде каталогов. Это Linux эквивалент DOS'овской утилиты zip, поддерживающая сохранение структуры каталогов в архиве.

5. Если вы уже имеете установленные исходники Linux, но хотите установить новое ядро, то вы должны удалить старые исходники. Используйте команду `rm -rf /usr/src/linux`
6. Теперь разожмите и извлеките исходники, используя команду `tar xzf linux-2.0.XX.tar.gz`
7. Теперь, `cd /usr/src/linux` и прочитайте файл README. Он содержит превосходное объяснение о том как настроить и скомпилировать новое ядро. Прочитайте этот файл (хорошая идея напечатать его и иметь копию под руками, пока вы компилируете, если у вас не было достаточно времени знать все по этому вопросу).

7.2 Знание ваших аппаратных средств

Вы ДОЛЖНЫ знать то, какие платы/устройства имеются внутри вашего PC, если вы собираетесь перекомпилировать ваше ядро!!! Для некоторых устройств (типа звуковых плат) вы должны также узнать различные установки (типа IRQ, адресов ввода-вывода и пр.).

7.3 Компиляция ядра - ядро Linux 1.2.13

Чтобы начать процесс конфигурации, следуйте инструкциям файла README, чтобы правильно установить исходники. Вы начинаете процесс настройки ядра командой `make config`

Для того, чтобы использовать PPP, вы должны сконфигурировать ядро, чтобы включить поддержку PPP (PPP требует И `pppd`, И поддержки PPP в ядре).

```
PPP (point-to-point) support (CONFIG_PPP) [n] y
```

Ответьте на другие вопросы `make config`, чтобы согласовать конфигурацию Linux с аппаратными средствами вашего PC, как вы хотите. Затем продолжите следовать README, чтобы скомпилировать и установить ваше новое ядро.

Ядро 1.2.13 создает только 4 PPP устройства. Для многопортовой последовательной платы, вы должны будете отредактировать исходники PPP ядра, чтобы получить большее количество портов. (См. README.LINUX файл, который приходит как часть дистрибутива PPP-2.1.2 для подробностей по правке исходных текстов, которую вы должны сделать).

Обратите внимание: диалог настройки 1.2.13 не позволяет вам идти в обратном направлении - так если вы допустили ошибку в ответе на один из вопросов в диалоге `make config`, выйдите, нажав CTRL-C и начните снова.

7.4 Компиляция ядра - ядра Linux 1.3.x и 2.0.x

Для Linux 1.3x и 2.0.x процесс настройки аналогичен настройке ядра 1.2.13. Снова следуйте инструкциям файла README, чтобы правильно установить исходники. Вы начинаете процесс конфигурации ядра командой

```
make config
```

Однако, вы также можете выбрать

```
make menuconfig
```

Это команда запускает менюоснованную систему настройки с интерактивной справкой, которая позволяет вам двигаться в обратном направлении и вперед в процессе конфигурации.

Имеется также настойчиво рекомендуемый интерфейс настройки для X Window System

```
make xconfig
```

Вы можете вкомпилировать поддержку PPP непосредственно в ваше ядро или как загружаемый модуль.

Если вы используете PPP непостоянно, то рекомендуется компилировать PPP поддержку как загружаемый модуль. При использовании 'kernel'd, ваше ядро автоматически загрузит модуль(и), требуемый для обеспечения поддержки PPP, когда вы начинаете ваш процесс связи PPP. Это сохраняет ценное пространство ОЗУ: часть ядра не может быть выгружена из памяти, а загружаемые модули автоматически удаляются, если не используются.

Чтобы сделать это, вы должны разрешить возможность загрузки модулей:

```
Enable loadable module support (CONFIG_MODULES) [Y/n/?] y
```

Чтобы добавить поддержку PPP в ядро, ответьте на следующий вопрос:

```
PPP (point-to-point) support (CONFIG_PPP) [M/n/y/?]
```

Для загружаемого модуля PPP, ответьте M, иначе, для компиляции PPP в ядро, ответьте Y.

В отличие от ядра 1.2.13, ядро 2.0.x создает PPP устройства на лету по необходимости и не нужно править источники, чтобы увеличить число доступных PPP устройств.

7.5 Обратите внимание на PPP-2.2 и /proc/net/dev

Если Вы используете PPP-2.2, вы обнаружите, что побочный эффект создания PPP устройств 'на лету' в том, что никакие устройства не обнаруживаются, если вы посмотрите /proc/net, пока устройство не создано запуском pppd:

```
[hartr@archenland hartr]$ cat /proc/net/dev
Inter-|   Receive   |   Transmit
face |packets errs drop fifo frame|packets errs drop fifo colls carrier
  lo:  92792   0   0   0   0   92792   0   0   0   0   0   0
  eth0: 621737  13  13   0  23  501621   0   0   0 1309   0
```

Как только вы запустите одну (или больше) услуг ppp, вы увидите записи типа (от сервера ppp):

```
[root@kepler /root]# cat /proc/net/dev
Inter-|   Receive   |   Transmit
face |packets errs drop fifo frame|packets errs drop fifo colls carrier
lo:  428021    0    0    0    0    428021    0    0    0    0    0
eth0:4788257  648  648  319  650  1423836    0    0    0  4623    5
ppp0:   2103    3    3    0    0    2017    0    0    0    0    0
ppp1:  10008    0    0    0    0    8782    0    0    0    0    0
ppp2:   305    0    0    0    0    297    0    0    0    0    0
ppp3:   6720    7    7    0    0    7498    0    0    0    0    0
ppp4:  118231  725  725    0    0   117791    0    0    0    0    0
ppp5:   38915    5    5    0    0    28309    0    0    0    0    0
```

7.6 Общие настройки ядра для PPP

Если вы устанавливаете ваш Linux PC как PPP сервер, вы должны вкомпилировать в IP поддержку форвардинга. Это также необходимо, если вы хотите использовать Linux для связывания LAN вместе или вашей LAN с Интернет.

Если вы связываете LAN с Интернет (или связываете вместе две LAN), вы должны быть подумать о защите. Добавление поддержки IP файервола в ядро - наверняка ДОЛЖНО БЫТЬ!

Также вам это понадобится, если вы захотите использовать IP masquerade, чтобы соединить LAN, которая использует любое из вышеупомянутых 'неподключенных' адресов IP сетей.

Чтобы разрешить IP Masquerade и IP fire walling, вы ДОЛЖНЫ ответить ДА на первый вопрос в процессе конфигурации:

```
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL)?
```

Хотя это может звучать немного пугающе для новых пользователей, многие пользователи активно используют возможности IP Masquerade and IP fire walling в ядре Linux 2.0.XX без проблем.

Как только вы установили и перезагрузили ваше новое ядро, вы можете начать конфигурировать и проверять вашу связь(и) PPP.

8 Получение нужной информации о PPP сервере

Прежде, чем вы сможете установить PPP соединение с сервером, вы должны получить следующую информацию (от службы поддержки системных администраторов/пользователей PPP сервера):

номер(а) телефона(ов). Если вы звоните через офисную АТС, то вам также нужен префиксный номер, который позволит вам звонить наружу - часто это цифры (0) или (9).

сервер использует ДИНАМИЧЕСКИЕ или СТАТИЧЕСКИЕ IP адреса?

Если сервер использует СТАТИЧЕСКИЕ IP адреса, то вы должны узнать какой IP адрес использовать на вашем конце PPP соединения. Если ваш ISP обеспечивает вас подсетью допустимых IP адресов, вы должны будете узнать IP адреса, которые вы можете использовать и сетевую маску (netmask).

Большинство поставщиков услуг Интернет использует ДИНАМИЧЕСКИЕ IP адреса. Как упомянуто выше, это имеет некоторое влияние на то, какими услугами вы можете пользоваться.

Однако даже если вы используете СТАТИЧЕСКИЕ IP адрес, большинство PPP серверов никогда (по соображениям безопасности) не позволит клиенту определять IP адрес, поскольку это - риск взлома. Вы должны знать эту информацию!

Каковы IP адреса серверов доменных имен ISP?

Должны иметься по крайней мере два, хотя необходим только один.

Здесь может поджидать проблема. PPP установка в MS WINDOWS 95 позволяет устанавливать адрес DNS как часть процесса установления связи. Так что ваш ISP (или справочная служба корпорации) может сообщить вам, что вам не нужно знать IP адрес DNS сервера(ов).

Для Linux, вы нуждаетесь в адресе по крайней мере одного DNS. Linux реализация PPP не позволяет установку IP адреса DNS динамически во время соединения - и возможно никогда не позволит.

Обратите внимание: хотя Linux (как PPP клиент) не может принимать адрес DNS от сервера, это может, действуя как сервер, передать эту информацию клиентам, используя опцию `pppd dns-addr`. Сервер требует использования PAP/CHAP?

Если дело обстоит так, то вы должны знать ваши "id", и "secret" для использования при соединении. (Скорее всего они совпадают с вашими именем пользователя и паролем на вашем ISP).

Сервер автоматически запускает PPP, или вы должны выдать какие-то команды, чтобы запустить PPP на сервере после регистрации?

Если вы должны выдать команду, чтобы запустить PPP, то что это за команда?

Сервер Windows NT Microsoft и, если так, он использует систему MS PAP/CHAP?

Многие корпоративные LAN, кажется, используют этот метод усиления защиты MS WINDOWS NT

Тщательно обратите внимание на эту информацию - вы собираетесь ее использовать!

9 Конфигурирование вашего модема и последовательного порта

Вы должны удостовериться, что ваш модем правильно установлен и что вы знаете, с каким последовательным портом он соединен.

Помните:

```
DOS com1: = Linux /dev/cua0 (и /dev/ttyS0)
DOS com2: = Linux /dev/cua1 (и /dev/ttyS1)
```

и т.д.

Также стоит запомнить, что, если вы имеете 4 последовательных порта, то на стандартном PC com1 и com3 совместно используют IRQ4, а com2 и com4 совместно используют IRQ3.

Если вы имеете устройства на стандартных последовательных портах, которые совместно используют IRQ с вашим модемом, то у вас будут проблемы. Вы должны удостовериться, что ваш последовательный порт модема находится на собственном, уникальном IRQ. Многие современных последовательных плат (и последовательные порты на материнской плате лучшего качества) позволяют вам менять IRQ последовательных портов.

Если у вас ядро Linux 2, то вы можете проверить "задействованные" IRQ, используя команду `cat /proc/interrupts`, которая выведет примерно вот что:

```
0:    6766283  timer
1:     91545  keyboard
2:         0  cascade
4:    156944 + serial
7:    101764  WD8013
10:   134365 + BusLogic BT-958
13:         1  math error
15:   3671702 + serial
```

Видно, что последовательные порты занимают IRQ4 (мышь) и IRQ15 (постоянная модемная PPP связь с Интернетом). (Имеется также последовательный порт на com2, IRQ3 и com4 находится на IRQ14, но поскольку они незадействованы, то они не обнаружены).

Предупреждение: Вы должны знать, что вы делаете, если вы собираетесь поиграться с вашими IRQ! Вы должны не только открыть ваш компьютер, вытащить платы и поиграть с джамперами,

но и должны знать какие IRQ используются для каждого устройства. В моем случае, компьютер оборудован только SCSI внешними устройствами, так что я могу отключить на материнской плате IDE интерфейсы, которые обычно используют IRQ14 и 15!

Также не забудьте, что, если на вашем PC загружаются другие операционные системы, изменение IRQ может привести к тому, что OS не сможет загружаться правильно - или вообще не загрузится! Если вы назначаете вашему последовательному порту не обычное IRQ, то вы должны сообщить Linux, какое IRQ какой порт использует. Это делается программой `setserial` и лучше всего выполнять это как часть процесса начальной загрузки в `rc.local` или `rc.serial`, который вызывается из `rc.local`, или как часть инициализации `SysV`. Для машины, рассмотренной выше, используются команды

```
/bin/setserial -b /dev/ttyS2 IRQ 11
/bin/setserial -b /dev/ttyS3 IRQ 15
```

Однако, если вы используете динамически загружаемые по требованию процесса `kernel` модули `serial`, то вы не можете установить один раз и забыть IRQ и т.д. во время начальной загрузки. Это потому, что при выгрузке модуля `serial` Linux забудет про специальные установки.

Итак, если вы загружаете модуль `serial` по требованию, то вы должны будете переконфигурировать IRQ и т.д. при каждой загрузке модуля.

9.1 Примечание, касающееся последовательных портов и возможностей быстрого действия

Если вы используете высокоскоростной (внешний) модем (14,400 кбит или выше), то ваш последовательный порт должен успевать обрабатывать трафик, причем модем особенно производителен при использовании алгоритмов сжатия данных.

Для этого требуется, чтобы ваш последовательный порт использовал современный UART (Универсальный Асинхронный Приемопередатчик) типа 16550(A). Если вы используете старую машину (или старую последовательную плату), то возможно, что ваш последовательный порт имеет только 8250 UART, который может вызвать у вас значительные проблемы при использовании с высокоскоростным модемом.

Используйте команду

```
setserial -a /dev/ttySx
```

чтобы Linux вывел вам тип UART, который у вас установлен. Если у вас нет UART типа 16550A, то купите новую последовательную плату (она стоит менее \$50). Когда вы приобретаете новую плату, удостоверьтесь, что вы можете на ней менять IRQ!

Обратите внимание: первые версии микросхем 16550 UART имели ошибку. Это было быстро обнаружено, и был выпущен исправленный вариант микросхемы - 16550A UART. Относительно небольшое число дефектных микросхем однако вошло в обращение. Маловероятно, что вы столкнетесь с одной из них, но вы должны посмотреть, точно ли у вас стоит 16550A, особенно на последовательных платах некоторого года изготовления.

9.2 Имена последовательных портов

Исторически, Linux использовал для исходящих звонков устройства `cuax` и устройства `ttySx` для входящих звонков.

Код ядра, который требовал это был изменен в версиях ядер 2.0.x, и вы должны теперь использовать `ttySx` и для исходящих, и для входящих звонков. Я полагаю, что имена устройств `cuax` могут совсем исчезнуть в будущих версиях ядра.

9.3 Конфигурирование вашего модема

Для PPP вы должны будете сконфигурировать ваш модем правильно - чтобы сделать это ПРОЧТИТЕ РУКОВОДСТВО ПО ВАШЕМУ МОДЕМУ! Большинство модемов приходит с фабричной настройкой по умолчанию, в которой определены опции, требуемые для PPP. Минимальная конфигурация определяет:

- Аппаратное управление потоком данных (RTS/CTS) (&K3 на многих Hayes модемах)
Вы должны проверить другие установки (в стандартных командах Hayes):
- E1 Command/usr/src/linux-2.0.27/include/linux/serial.h Эхоотображение ВКЛ (требуемый для работы chat)
- Q0 Отчет по кодам результата (требуемый для работы chat)
- S0=0 Автоответ ВЫКЛ (если вы не хотите, чтобы модем брал трубку)
- &C1 Обнаружение несущей (Carrier Detect) ВКЛ только после соединения
- &S0 Готовность данных (DSR) всегда ВКЛ
- (в зависимости) Data Terminal Ready

Имеется сайт, предлагающий установки модемов для разнообразных типов модемов,, который может помочь вам в этом.

Также неплохо бы исследовать, как функционирует последовательный интерфейс между вашим компьютером и модемом. Наиболее современные модемы позволяют фиксировать скорость последовательного интерфейса, позволяя изменять скорость на интерфейсе телефонной линии до наивысшего возможного (для взаимодействия локального и удаленного модема) значения.

Это известно как операция разбиения скорости. Если ваш модем поддерживает ее, то заблокируйте последовательный интерфейс модема на самой высокой доступной скорости (обычно 115,200 бод, но возможно и 38,400 бод для модемов 14,400).

Используйте ваше коммуникационное программное обеспечение (например minicom или seyon), чтобы выяснить конфигурацию вашего модема, и установите его для PPP.

Многие модемы сообщают свои текущие установки в ответ на AT\$V, но вы должны свериться с вашим руководством по модему.

Если вы полностью сбили установки, то вы (обычно) можете возвратиться к первоначальным, дав команду AT\$F - возвращение к установкам. (Для большинства модемов, с которыми я сталкивался, фабричные установки включают все необходимое для PPP - но все-таки вы должны это проверить). Как только вы составили строку установки модема, требуется вписать ее. Примите решение: вы можете сохранить эти установки в энергонезависимой памяти вашего модема, так что они могут быть задействованы соответствующей командой AT. В качестве альтернативы вы можете передавать правильные установки на ваш модем как часть процесса установки соединения PPP.

Если вы используете ваш модем только чтобы звонить из Linux вашему ISP или на корпоративный сервер, то самым простым для вас будет сохранить настройки вашего модема в энергонезависимом ОЗУ.

Если, с другой стороны, ваш модем используется другими приложениями и операционными системами, то безопаснее передавать эту информацию на модем, поскольку при каждом звонке предполагается, что модем гарантированно находится в правильном состоянии. (Этот подход имеет дополнительное преимущество, выраженное в том, что не будет страшно потерять строку установки модема, записанную в энергонезависимом ОЗУ, что может действительно случиться).

9.4 Обратите внимание на управление потоком данных последовательного интерфейса

Когда данные путешествуют по последовательным линиям связи, может случаться так, что данные прибывают быстрее, чем компьютер может их обработать (компьютер может быть

занят выполнением чего-то еще - помните, что Linux - многопользовательская многозадачная операционная система). Чтобы гарантировать, что данные не потерялись (входной буфер может переполниться и следовательно данные теряются), нужен какой-то способ управления потоком данных.

- На последовательных линиях есть два главных способа это сделать:
- Использование аппаратных сигналов (Clear To Send/Request to Send - CTS/RTS)
- Использование программных сигналов (Control-S и Control-Q, также известные как XON/XOFF).

В то время как последний может хорошо подходить для терминальной (текстовой) связи, данные при PPP связи использует все 8 битов - и очень вероятно, что где-нибудь в данных появятся байты, которые могут быть истолкованы как Control-S и Control-Q. Так что, если модем установлен с использованием программного управления потоком данных, то things can rapidly go berserk!

Для высокоскоростных связей, использующих PPP (который использует 8 битов данных) необходимо аппаратное управление потоком данных, и по этой причине вы должны использовать аппаратное управление потоком данных.

9.5 Тестирование вашего модема на исходящие звонки

Теперь, когда вы отобрали наружный последовательный порт и установили модем, хорошо бы удостовериться, что эти установки действительно работают, позвонив вашему ISP и таким образом выяснить, можете ли вы с ним соединиться.

При использовании пакета вашего коммуникационного терминала (типа minicom), установите требуемую для PPP инициализацию модема, и позвоните на PPP сервер, с которым вы хотите осуществить сеанс PPP связи.

(Обратите внимание: в этой стадии мы не пробуем создавать PPP соединение - только хотим убедиться, что мы имеем правильный номер телефона и также выяснить что именно сервер посылает нам для регистрации в системе и запуска PPP).

В течение этого процесса, или "захватите"(направьте лог в файл) весь процесс входа в систему в файл или тщательно (очень тщательно) запишите в точности, какие подсказки сервер выдает вам при вводе вашего имени пользователя и пароля (и любые другие команды, необходимые для установления PPP соединения).

Если ваш сервер использует PAP, то вы не должны видеть подсказку входа в систему, но должны взамен видеть на вашем экране (в текстовом представлении) запуск протокола управления связи (который похож на мусор).

Несколько предупреждений:

некоторые серверы очень интеллектуальны: вы можете регистрироваться либо используя текстовые приглашения ко вводу имени/пароля пользователя, либо используя PAP. Так что, если ваш ISP или корпоративный сайт использует PAP, но вы не видите тут же появляющийся мусор, то это не обязательно означает, что вы сделали что-то неправильно.

- некоторые серверы требуют, чтобы вы ввели сначала некоторый текст и только потом начинают посылать стандартные PAP последовательности.
- Некоторые PPP серверы пассивны - который ничего не делают до тех пор, пока клиент не отправит им допустимый lcr пакет. Если ppp сервер, с которым вы соединяетесь, функционирует в пассивном режиме, то мусор вы не увидите!
- Некоторые серверы не запускают PPP, пока вы не нажмете enter - так что, если вы правильно регистрируетесь и не видите мусор, то попробуйте нажать enter!

Стоит позвонить по крайней мере дважды - некоторые серверы изменяют свои подсказки (например, со временем!) при каждой последующей регистрации. Две критические подсказки, которые ваш Linux компьютер должен уметь определять:

- подсказка, которая требует, чтобы вы ввели ваше имя пользователя;
- подсказка, которая требует, чтобы вы ввели ваш пароль;

Если вы должны выдать команду, чтобы запустить PPP на сервере, то вы будете также должны выяснить, какую подсказку сервер выдает вам, для ввода этой команды.

Если ваш сервер автоматически запускает PPP, как только вы зарегистрировались, то вы увидите мусор на вашем экране - это PPP сервер посылает на вашу машину информацию, чтобы запустить и сконфигурировать PPP соединение.

Это должно выглядеть примерно так

```
~y}#.!!}!!} }8}!}$}%U}"& } } } }%}& ...}'"}{ }" } .~y}
```

(and it just keeps on coming!)

На некоторых системах PPP должен быть явно запущен на сервере. Это обычно из-за того, что сервер был установлен с позволением входить по PPP-логину в систему и запускать системный шелл, используя ту же самую пару имени/пароля пользователя. Если дело обстоит так, то выдайте эту команду, когда зарегистрируетесь в системе. Снова вы увидите мусор, так как серверная сторона запустила PPP соединение.

Если вы не видите это сразу же после соединения (и регистрации в системе и запуска PPP сервера, если требуется), то нажмите enter, чтобы увидеть, запустится ли PPP сервер ...

В этой точке, вы можете сказать вашему модему "положить трубку" (обычно, для этого надо быстро напечатать + + + и потом дать команду ATHO, если ваш модем ответил ОК).

Если вы не можете заставить ваш модем работать, читать руководство по вашему модему, man по вашей коммуникационной программе и Serial HOWTO! Когда вы их изучите, действуйте как сказано выше.

10 Настройка преобразования доменных имен в IP адрес (DNS)

Хотя нам - людям нравится давать названия вещам, компьютеры в действительности больше любят числа. В TCP/IP сетях (которые составляют Интернет), мы называем машину определенным именем - и каждая машина живет в определенной области - "домене". Например, моя рабочая станция Linux называется archenland, и постоянно находится в домене interweft.com.au . Читательный для человека адрес таким образом - archenland.interweft.com.au (такая запись известна также как FQDN - полностью квалифицированное имя домена).

Однако в Интернет машины, общаясь между собой, вместо имен используют IP адреса.

Трансляция имени машины (и домена) в адрес, фактически используемый в Интернет, - это задача машин, которые предоставляют Domain Name Service (обслуживание доменных имен).

Как это происходит:

- ваша машина должна знать IP адрес какого-то компьютера.
Приложение, требующее эту информацию запрашивает 'resolver' на вашем Linux PC, чтобы предоставить эту информацию;
- resolver делает запрос к локальному файлу хостов (/etc/hosts и/или серверу доменных имен, которые могут дать такую информацию (точнее поведение resolver'a определено в /etc/host.conf);
- если ответ найден в файле хостов, то ответ возвращается;
- если определен сервер доменных имен, то ваш PC делает запрос этой машине;

- если DNS машина уже знает IP адрес для требуемого имени, то она возвращает его. Если нет, то делает запрос другим серверам доменных имен в Интернет, чтобы найти информацию. Сервер доменных имен затем посылает эту информацию обратно на запрашивающий resolver - который в свою очередь передает информацию запросившему приложению.

Когда вы устанавливаете PPP соединение, вы должны сообщить вашей Linux машине, где она может получить информацию по имени хоста по IP адресу (разрешение адреса) так, чтобы вы могли использовать имена машин, а ваш компьютер мог транслировать их в IP адреса, которые требуются для работы.

Один из способов состоит в том, чтобы прописать каждый хост, с которым вы хотите общаться в файл /etc/hosts (что нереально, если вы соединяетесь с Интернет); другой - использовать машинный IP адрес вместо имени (тоже невозможно, кроме самых маленьких LAN).

Самый лучший способ состоит в том, чтобы установить Linux так, чтобы он мог автоматически получить информацию об адресе по имени. Эту услугу предоставляет система сервера доменных имен (DNS). Все, что необходимо, это ввести IP адрес(а) серверов доменных имен в ваш файл /etc/resolv.conf .

10.1 Файл /etc/resolv.conf

Служба поддержки пользователей вашего PPP сервера должна обеспечить вас IP адресами DNS (необходим только один - но два адреса дают некоторую избыточность в случае отказа одного из них).

Как упоминалось выше, Linux не может устанавливать IP адрес сервера имен тем же способом, что и MS WINDOWS 95. Так что вы должны настоять (вежливо), чтобы ваш ISP обеспечил вас этой информацией!

Ваш /etc/resolv.conf должен выглядеть примерно так:

```
domain your.isp.domain.name
nameserver 10.25.0.1
nameserver 10.25.1.2
```

Отредактируйте этот файл (создайте в случае необходимости) так, чтобы прописать информацию, которую дал ваш ISP. Он должен иметь следующие права владения и доступа:

```
-rw-r--r--  1 root    root          73 Feb 19 01:46 /etc/resolv.conf
```

Если вы уже настроили /etc/resolv.conf, потому что вы находитесь в LAN, просто добавьте IP адреса PPP DNS серверов к вашему существующему файлу.

10.2 Файл /etc/host.conf

Вы должны также проверить, что ваш файл /etc/host.conf правильно настроен. Он должен выглядеть примерно так

```
order hosts,bind
multi on
```

Эти строки сообщают resolver'у, что сперва надо искать информацию в файле хостов, а потом уже посылать DNS запросы.

11 Использование PPP и привилегии root

Поскольку PPP должен настроить сетевые устройства, измените таблицу маршрутизации ядра и т.д, что требует привилегий root.

Если пользователи (не root) должны устанавливать PPP соединение, то программа pppd должна были setuid root:

```
-rwsr-xr-x  1 root      root          95225 Jul 11 00:27  /usr/sbin/pppd
```

Если /usr/sbin/pppd не setuid, то зарегистрируйтесь как root и дайте команду:

```
chmod u+s /usr/sbin/pppd
```

После этого pppd бкдет запускаться с привилегиями root, даже если запущен обычным пользователем. Это позволяет нормальному пользователю выполнять pppd с необходимыми привилегиями, чтобы настроить сетевые интерфейсы и таблицу маршрутизации ядра.

Программы, которые выполняют 'set uid root', являются потенциальными лазейками в защите системы, и вы должны быть чрезвычайно осторожны при создании программ 'suid root'. Ряд программ (включая pppd) был тщательно написан с точки зрения минимизации опасности выполнения suid root, так что вы должны быть защищены (но все-таки гарантий нет).

В зависимости от того, как вы хотите, чтобы ваша система работала - особенно если вы хотите, чтобы ЛЮБОЙ пользователь вашей системы мог инициализировать PPP связь, то вы должны сделать ваш набор скриптов ppp-on/off читаемым/выполнимым. (Наверное это здорово, когда РС используется ТОЛЬКО Вами).

Однако, если вы НЕ хотите, чтобы любой человек мог запускать PPP соединение (например, ваши дети имеет аккаунты на вашем Linux PC, и вы не хотите, чтобы они выходили в Интернет без вашего разрешения), то вы должны будете настроить группу PPP (как root, отредактировать /etc/group) и:

- сделайте для файла pppd - suid root, владелец root и группа PPP, с очищенными правами 'other'. Это должно затем походить

```
-rwsr-x---  1 root      PPP          95225 Jul 11 00:27  /usr/sbin/pppd
```

- сделайте для скриптов ppp-on/off - владелец root и группа PPP
- сделайте скрипты ppp-on/off читаемыми/выполнимыми группой PPP

```
-rwxr-x---  1 root      PPP          587 Mar 14  1995  /usr/sbin/ppp-on
-rwxr-x---  1 root      PPP          631 Mar 14  1995  /usr/sbin/ppp-off
```

- обнулите права other для ppp-on/off.
- добавьте пользователей, которые будут запускать PPP в группу PPP в /etc/group

Даже если вы сделали это, обычные пользователи ВСЕ ЕЩЕ не могут завершить связь программно! Выполнение сценария ppp-off требует привилегий root. Однако, любой пользователь может только выключить модем (или выдернуть телефонную линию из внутреннего модема).

Вариант (и лучший метод), чтобы это настроить, состоит в использовании программы sudo. Она предлагает повышенную защиту и позволит вам настроить систему так, чтобы любой (разрешенный) пользователь мог активизировать/дезактивировать связь, используя скрипты. Использование sudo позволит разрешенному пользователю активизировать/дезактивировать связь PPP чисто и надежно.

12 Настройка файлов PPP соединения

Вы теперь должны зарегистрироваться как `root`, чтобы создать каталоги и отредактировать файлы, необходимые для установки PPP, даже если вы хотите, чтобы PPP был доступен для всех пользователей.

PPP использует ряд файлов для соединения и установки PPP соединения.

Они отличаются по имени и расположению между версиями PPP 2.1.2 и 2.2.

Для PPP 2.1.2 файлы:

```

/usr/sbin/pppd           # the PPP binary
/usr/sbin/ppp-on        # the dialer/connection script
/usr/sbin/ppp-off       # the disconnection script
/etc/ppp/options        # the options pppd uses for all connections
/etc/ppp/options.ttyXX  # the options specific to a connection on this port

```

Для PPP 2.2 файлы:

```

/usr/sbin/pppd           # the PPP binary
/etc/ppp/scripts/ppp-on  # the dialer/connection script
/etc/ppp/scripts/ppp-on-dialer # part 1 of the dialer script
/etc/ppp/scripts/ppp-off # the actual chat script itself
/etc/ppp/options        # the options pppd uses for all connections
/etc/ppp/options.ttyXX  # the options specific to a connection on this port

```

Пользователи Red Hat Linux должны обратить внимание на то, что во время стандартной установки Red Hat 4.X эти скрипты находятся в `/usr/doc/ppp-2.2.0f-2/scripts`.

В вашем каталоге `/etc` должен быть каталог `ppp`:

```
drwxrwxr-x  2 root  root      1024 Oct  9 11:01 ppp
```

Если он не существует - создайте его с этими правами владения и доступа.

Если каталог уже существовал, он должен содержать шаблон файла `options`, называемый `options.tpl`. Этот файл включен ниже на тот случай, если у вас его нет.

Напечатайте его, поскольку он содержит объяснение почти всех PPP опций (их полезно читать вместе с `man pppd`). Хотя вы можете использовать этот файл как ваш базовый файл `/etc/ppp/options`, вероятно лучше создать ваш собственный файл `options`, который не будет включать комментарии из шаблона, тогда он будет намного короче и более легок для чтения/сопровождения.

Если вы имеете несколько последовательных линий/модемов (обычный случай для PPP сервера), создайте общий файл `/etc/ppp/options`, содержащий опции, которые являются общими для всех последовательных портов, на которых вы осуществляете входящие/исходящие звонки и создайте индивидуальные файлы `options` для каждой последовательной линии, на которой вы будете устанавливать PPP соединение с индивидуальными установками, требуемыми для каждого порта.

Эти файлы со специфическими для портов опциями именуются `options.ttyx1`, `options.ttyx2` и т.д (где `x` - соответствующий символ для ваших последовательных портов).

Однако, для одиночного PPP соединения, вы можете вполне спокойно использовать файл `/etc/ppp/options`. В качестве альтернативы вы можете поместить все опции как параметры непосредственно в команде `pppd`.

Проще сопровождать настройку, которая использует файлы `/etc/ppp/options.ttySx`. Если вы используете PPP, чтобы соединиться с рядом различных мест, то вы можете создать файлы `options` для каждого места в `"/etc/ppp/options.место"` и затем определять файл опций как параметр для команды PPP при соединении (используя опцию `pppd "file option-file"` в командной строке).

12.1 Файл options.tpl

Некоторые дистрибутивы PPP, кажется, потеряли файл options.tpl, так что вот полный файл. Я предлагаю вам не редактировать этот файл, а создать ваш(и) файл(ы) /etc/ppp/options. Достаточно скопировать его в новый файл и затем отредактировать. Если вы запутались в ваших ипращениях, то вы можете затем вернуться к оригиналу и начать снова.

```
# /etc/ppp/options -*- sh -*- общие опции для pppd
# создано 13-Jul-1995 jmk
# автоматически проставленная дата: 01-Aug-1995
# автоматически проставленное время: 19:45

# Использовать указанную команду или программу для настройки линии. Этот
# обычно использует программу "chat", чтобы позвонить модемом и запустить
# удаленную ppp сессию.
#connect "echo Вам нужно установить команду connect."

# Запустить указанную команду или программу, когда pppd завершил связь.
# Этот скрипт мог бы, к примеру, ввести команды в модем, чтобы завершить
# связь, если сигналы аппаратного управления модемом недоступны
#disconnect "chat -- \d+++ \d\c OK ath0 OK"

# async карта символов -- 32-bit hex; каждый бит - символ, который надо
# представить в виде escape-последовательности, чтобы pppd мог его принять.
# 0x00000001 - это маска для '\x01', а 0x80000000 - маска для '\x1f'.
#asynsmap 0

# Требование для удаленной стороны назвать себя перед тем как начнется обмен
# пакетами.
#auth

# Использовать аппаратное управление потоком данных (напр., RTS/CTS), чтобы
# управлять потоком данных на последовательном порту
#crtcts

# Использовать программное управление потоком данных (напр., XON/XOFF), чтобы
# управлять потоком данных на последовательном порту
#xonxoff

# Добавить маршрут по умолчанию в системную таблицу маршрутизации,
# используя удаленную сторону как шлюз, когда договоренность IPCP успешно
# достигнута. Эта запись удаляется при завершении связи
#defaultroute

# Определяет, что некоторые символы должны быть за'escape'лены при передаче
# (Независимо от того, просила ли удаленная сторона их escape'ить своей
# async картой управляющих символов). Символы, которые надо escape'ить
# указываются в виде списка шестнадцатеричных чисел, разделенных запятыми.
# Заметьте, что для опции escape могут быть указаны почти все символы, в
# отличие от опции asynsmap, которая позволяет указывать только управляющие
# символы. Символы, которых нельзя escape'ить - это 0x20 - 0x3f или 0x5e.
#escape 11,13,ff
```

```
# не использовать линии управления модемом.
#local

# Указывает, что pppd должен использовать lock в стиле UUCP на
# последовательное устройство, чтобы исключить одновременный доступ к
# устройству
#lock

# Использовать линии управления модемом. На Ultrix эта опция означает
# управление потоком данных, как опция crtscts. (Эта опция не полностью
# выполняется)
#modem

# Установить значение MRU [Maximum Receive Unit] в <n> при договоренности.
# pppd запросит удаленную сторону отправлять пакеты не более, чем по <n> байт.
# Минимальное значение MRU 128. Значение MRU по умолчанию 1500. Для
# медленных линков рекомендуется 296 (40 байт для заголовка TCP/IP + 256 байт
# данных).
#mru 542

# Установить сетевую маску интерфейса в <n>, 32-битная сетевая маска в
# "десятично-точечной" нотации (напр., 255.255.255.0).
#netmask 255.255.255.0

# Запретить поведение по умолчанию, когда не указан локальный IP адрес,
# которое определяет (если возможно) локальный IP адрес по имени хоста
# С этой опцией удаленная сторона должна будет обеспечить локальный IP
# адрес в течение IPCP переговоров (если она не определена явно в командной
# строке или в файле options).
#noipdefault

# Разрешить опцию "passive" в LCP. С этой опцией pppd будет пытаться
# инициировать соединение; если ответ от другой стороны не принят, то
# pppd будет пассивно ожидать правильный LCP пакет от другой стороны
# (вместо выхода, как делается без этой опции).
#passive

# С этой опцией pppd не будет передавать LCP пакеты для инициации соединения
# пока не придет правильный LCP пакет от другой стороны (как опция "passive"
# в старых версиях pppd).
#silent

# Не требовать или не разрешать договариваться о любых опциях LCP и IPCP
# (использовать значения по умолчанию).
#-all

# Запретить договариваться о сжатии адреса/управления (используется
# по умолчанию, т.е. адресные/управляющие поля запрещены).
#-ac

# Запретить договариваться о asynstar (используется asynstar по умолчанию, т.е.
# escape все управляющие символы).
#-am
```



```
# Не переходить в фоновый режим (иначе rpprd бдет делать так, если
# последовательной устройство указано).
#-detach

# Запретить договариваться о IP адресе (С этой опцией удаленный IP
# адрес должен указываться опцией командной строки или в файле options)
#-ip

# Запретить договариваться о магических числах. С этой опцией rpprd не может
# определить looped-back линию.
#-mn

# Запретить договариваться о MRU [Maximum Receive Unit]
# (используется по умолчанию, т.е. 1500).
#-mru

# Запретить договариваться о сжатии полей протокола (используется по
# умолчанию, т.е. сжатие полей протокола запрещено).
#-pc

# Требуется, чтобы другая сторона назвала себя используя PAP.
# Это требует ДВУНАПРАВЛЕННОЙ аутентификации - НЕ используйте эту
# опцию для стандартной аутентификации PAP на ISP, так как при этом требуется,
# чтобы машина ISP аутентифицировала себя на вашей машине (а она это не сможет
# сделать)
#+pap

# Не соглашаться с аутентификацией PAP.
#-pap

# Требуется, чтобы другая сторона назвала себя используя CHAP [Cryptographic
# Handshake Authentication Protocol].
# Это требует ДВУНАПРАВЛЕННОЙ аутентификации - НЕ используйте эту опцию
# для стандартной аутентификации CHAP на ISP, так как при этом требуется,
# чтобы машина ISP аутентифицировала себя на вашей машине (а она это не сможет
# сделать)
#+chap

# Не соглашаться с аутентификацией CHAP.
#-chap

# Запретить обсуждение сжатия IP заголовков в стиле Van Jacobson
# (используется по умолчанию, т.е. сжатия нет)
#-vj

# Увеличить уровень отладки (то же что -d). Если эта опция есть, rpprd
# будет записывать в журнал все прибывшие и отправленные пакеты в читабельной
# форме. Пакеты регистрируются в лог-файлах через syslog со средством daemon
# и уровнем отладки. Эта информация может быть перенаправлена в файл
# соответствующей установкой /etc/syslog.conf (см. syslog.conf(5)). (Если
# rpprd скомпилирован с разрешенной extra отладкой, он будет записывать
# сообщения в журнал, используя средство local2 вместо daemon).
```

```
#debug

# Добавить имя домена <d> к локальному имени хоста для целей аутентификации.
# Например, если gethostname() возвращает имя porsche, но FQDN -
# porsche.Quotron.COM, то вы должны установить эту опцию, чтобы имя домена
# было Quotron.COM.
#domain <d>

# Разрешить отладочный код в PPP драйвере на уровне ядра. Аргумент n
# это число-комбинация следующих значений: 1 - разрешить общие отладочные
# сообщения, 2 - запросить, надо ли печатать содержимое принимаемых пакетов
# и 4 - запросить, надо ли печатать содержимое отправляемых пакетов
#kdebug n

# Установить значение MTU [Maximum Transmit Unit] в <n>. Пока другая сторона
# на попросит меньшее значение при договоре о MRU, pppd будет требовать у
# сетевого кода ядра отправлять пакеты данных не более, чем по n байт
# через сетевой интерфейс PPP.
#mtu <n>

# Установить имя локальной системы для аутентификационных целей в <n>.
# Возможно, ее надо будет установить в ваше имя на вашем ISP,
# если используется PAP/CHAP.
#name <n>

# Установить имя пользователя для аутентификации этой машины на другой стороне
# используя PAP, в значение <u>.
# НЕ использовать, если вы использовали опцию 'name'!
#user <u>

# Заставить использовать имя хоста как имя локальной системы для
# аутентификационных целей (перекрывает опцию name).
#usehostname

# Установить имя удаленной системы для аутентификационных целей в <n>.
#remotename <n>

# Добавить запись для этой системы в таблицу ARP [Address Resolution Protocol]
# с IP адресом удаленной системы и Ethernet адресом этой системы
#proxyarp

# Использовать базу данных паролей для идентификации удаленной стороны
# используя PAP.
#login

# Если эта опция назначена, pppd будет отправлять кадр LCP echo-request
# удаленной стороне каждые n секунд. Под Linux, echo-request отправляется
# когда пакеты не принимаются от удаленной стороны n секунд. Обычно
# удаленная сторона должна отвечать на echo-request отправкой echo-reply.
# Эта опция может быть использована с опцией lcp-echo-failure для
# определения что удаленная сторона больше не соединена.
#lcp-echo-interval <n>
```

```
# Если эта опция задана, pppd будет считать, что связи с peer нет, если п
# LCP echo-requests отправлены без приема правильных LCP echo-reply.
# Если это случилось, pppd завершит связь. Использование этой опции
# требует ненулевого значения для параметра lcp-echo-interval.
# Эта опция может быть использована, чтобы разрешить pppd завершать связь,
# если физическое соединение отсутствует (напр, модем бросил трубку) в
# ситуациях, когда доступны линии управления модемом.
#lcp-echo-failure <n>

# Установить интервал рестарта LCP (пауза возобновления передачи) в <n> секунд
# (по умолчанию 3).
#lcp-restart <n>

# Установить максимальное число передач LCP terminate-request в <n>
# (по умолчанию 3).
#lcp-max-terminate <n>

# Установить максимальное число передач LCP configure-request в <n>
# (по умолчанию 10).
# Некоторые PPP сервера долго стартуют. Вам может понадобиться увеличить эту
# величину, если вы получаете ошибки 'serial line looped back' и вы УВЕРЕННЫ,
# что корректно регистрируетесь и PPP должен запускаться на сервере.
#lcp-max-configure <n>

# Установить максимальное число LCP configure-NAKs, возвращенных перед началом
# отправки вместо configure-Rejects, в <n> (по умолчанию 10).
#lcp-max-failure <n>

# Установить интервал рестарта IPCP (пауза возобновления передачи) в <n>
# секунд (по умолчанию 3).
#ipcp-restart <n>

# Установить максимальное число передач IPCP terminate-request в <n>
# (по умолчанию 3).
#ipcp-max-terminate <n>

# Установить максимальное число передач IPCP configure-request в <n>
# (по умолчанию 10).
#ipcp-max-configure <n>

# Установить максимальное число IPCP configure-NAKs, возвращенных перед началом
# отправки вместо configure-Rejects, в <n> (по умолчанию 10).
#ipcp-max-failure <n>

# Установить интервал рестарта PAP (пауза возобновления передачи) в <n> секунд
# (по умолчанию 3).
#pap-restart <n>

# Установить максимальное число передач PAP authenticate-request в
# <n> (по умолчанию 10).
#pap-max-authreq <n>

# Установить интервал рестарта CHAP (пауза возобновления передач
```

```

# challenges) в <n> секунд (по умолчанию 3).
#chap-restart <n>

# Установить максимальное число передач CHAP challenge в <n>
# (по умолчанию 10).
#chap-max-challenge

# С этой опцией pppd будет re-challenge удаленную сторону каждые <n>
# секунд.
#chap-interval <n>

# С этой опцией pppd будет принимать предложение peer о нашем локальном IP
# адресе, даже если локальный IP адрес был указан в опциях.
#ipcp-accept-local

# С этой опцией pppd будет принимать предложение peer о его (удаленном) IP
# адресе, даже если удаленный IP адрес был указан в опциях.
#ipcp-accept-remote

```

12.2 Какие опции я должен использовать? (PAP/CHAP нет)

Ну, как и во всех вещах, это зависит от многого (вздых). Опции, определенные здесь должны работать с большинством серверов.

Однако, если они не работает, ЧИТАЙТЕ ФАЙЛ-ШАБЛОН (/etc/ppp/options.tpl) и man pppd, и поговорите со службой поддержки сервера, к которому вы подсоединяетесь.

Вы должны также обратить внимание, что скрипты соединения, представленные здесь также используют некоторые опции командной строки для pppd, что позволяет более просто вносить изменения

```

# /etc/ppp/options (NO PAP/CHAP)
#
# Чтобы pppd не уходил в фоновый режим
-detach
#
# использовать линии управления модемом
modem
# использовать залочку портов в стиле uucsr, чтобы одновременно несколько
# последовательных устройств не обращались к одному порту
lock
# использовать аппаратное управление потоком данных
crtstcts
# сделать это соединение маршрутом по умолчанию в таблице маршрутизации
defaultroute
# НИЧЕГО не "escape-ить"
asynsmar 0
# использовать максмальный размер передаваемого пакета в 552 байт
mtu 552
# использовать максмальный размер принимаемого пакета в 552 байт
mru 552
#
#-----END OF SAMPLE /etc/ppp/options (no PAP/CHAP)

```

13 Если ваш PPP сервер использует PAP (Password Authentication Protocol)

Если сервер, с которым вы соединяетесь, требует PAP или CHAP, то вам придется поработать несколько больше.

К вышеупомянутому файлу options добавьте следующие строки

```
#
# заставить pppd использовать ваше ISP имя пользователя в качестве вашего
# 'имени хоста' при аутентификации
name <your ISP user name>      # исправьте эту строку как вам нужно
#
# Если вы запустили PPP *сервер* и непременно нужен PAP или CHAP
# раскомментируйте соответственно одну из этих строк. НЕ используйте их,
# если вы - клиент, подключающийся к PPP серверу (даже если он использует PAP
# или CHAP), так как это говорит СЕРВЕРУ аутентифицировать себя на вашей
# машине (что почти никогда нельзя сделать - и связь не будет установлена).
#+chap
#+pap
#
# Если вы используете ЗАШИФРОВАННЫЕ сведения в файле /etc/ppp/pap-secrets,
# то раскомментируйте следующую строку.
# Заметьте: это НЕ тоже, что использование MS шифрации паролей, которая
# может стоять на MS RAS Windows NT.
#+papcrypt
```

13.1 Использование MSCHAP

Windows NT Microsoft RAS может использовать разновидность CHAP. В вашем дистрибутиве исходных текстов PPP, вы найдете файл README.MSCHAP80, который обсуждает этот вопрос.

Вы можете определить, запрашивает ли сервер установление подлинности, используя этот протокол, разрешив отладку для pppd. Если сервер запрашивает установление подлинности по MS CHAP, то вы увидите строки типа

```
rcvd [LCP ConfReq id=0x2 <asynmap 0x0> <auth chap 80> <magic 0x46a3>]
```

Критическая информация здесь - auth chap 80.

Для того, чтобы использовать MS CHAP, вы должны будете пересобрать pppd, чтобы вкомпилировать эту поддержку. Пожалуйста, см. инструкции в файле README.MSCHAP80 в исходных текстах PPP, чтобы узнать, как вкомпилировать и использовать этот протокол.

Вы должны обратить внимание, что в настоящее время этот код в Linux поддерживает только клиент PPP, соединяющийся с сервером MS WINDOWS NT. Сервер PPP в Linux не поддерживает этот протокол для установления подлинности клиентов по протоколу MSCHAP80.

13.2 Файл секретов PAP/CHAP

Если вы используете установление подлинности по pap или chap, то вы также должны создать файл секретов. Это:

```
/etc/ppp/pap-secrets
/etc/ppp/chap-secrets
```

Для защиты они должны принадлежать пользователю root, группе root и иметь права доступа к файлу 740.

Первое, на что надо обратить внимание насчет PAP и CHAP - то, что они разработаны, чтобы опознавать компьютерные системы, а не пользователей.

"Оп-па!.. А какая разница?" слышу я ваш вопрос.

А такая что, как только ваш компьютер создал PPP соединение с сервером, ЛЮБОЙ пользователь вашей системы может использовать это соединение - не только вы. Именно поэтому вы можете устанавливать WAN связь, которая соединяет две LAN, используя PPP.

PAP может требовать (а CHAP ТРЕБУЕТ) двунаправленного установления подлинности, то есть компьютер на каждом конце соединения должен иметь правильные имя и пароль другой стороны. Однако, большинство PPP серверов с доступом по коммутируемым линиям, использующие PAP, этим способом НЕ пользуются.

Как уже упоминалось, ваш ISP вероятно даст вам имя пользователя и пароль, чтобы позволить вам соединиться с их системой и выходить дальше в Интернет. Ваш ISP не интересуется именем вашего компьютера вообще, так что вы вероятно будете должны использовать ваше ISP имя пользователя как имя вашего компьютера.

Это делается, используя имя пользователя в опции name rppd. Так, если вы должны использовать имя пользователя, выданное вам вашим ISP, добавьте строку

```
name your_user name_at_your_ISP
```

к вашему файлу /etc/ppp/options.

Технически, вы действительно должны использовать пользователя our_user name_at_your_ISP для PAP, но rppd достаточно интеллектен, чтобы интерпретировать имя как пользователя, если это требуется для использования PAP. Преимущество использования опции name в том, что она также является допустимой для CHAP.

Так как PAP служит для опознания компьютеров, технически вы должны также определить удаленное компьютерное имя. Однако, поскольку большинство людей работает только с одним ISP, то вы можете использовать групповой символ (*) для удаленного имени хоста в файле секретов.

Также стоит заметить, что многие ISP применяют модемные пуллы, соединенные с различными терминальными серверами - каждый со своим именем, но ДОСТУПНЫЙ с одного (циклически переключаемого) входного номера. Следовательно может быть очень трудно при некоторых обстоятельствах знать заранее имя удаленного компьютера, поскольку это зависит от того, к какому терминальному серверу вы подсоединились!

13.3 Файл секретов PAP

Файл /etc/ppp/pap-secrets выглядит примерно так:

```
# Secrets for authentication using PAP
# client      server      secret      acceptable_local_IP_addresses
```

Четыре поля - разграниченные пробельными символами, причем последнее может быть пустым (которое вы можете захотеть использовать для динамического и вероятно статического IP распределение от вашего ISP).

Предположим, что ваш ISP дал вам имя пользователя fred и пароль flintstone, установите опцию name fred в /etc/ppp/options[.ttySx] и настройте ваш файл /etc/ppp/pap-secrets следующим образом

```
# Secrets for authentication using PAP
# client      server  secret          acceptable local IP addresses
fred          *      flintstone
```

Обратите внимание, что мы не должны определять локальный адрес IP, если мы не требуем ПРИНУДИТЕЛЬНО выставить локальный, статический адрес IP. Даже если вы попытаете это сделать, вряд ли что-нибудь получится, так как большинство PPP серверов (для защиты) не позволяют удаленной системе устанавливая свой IP адрес, они должны быть выданы со стороны ISP.

13.4 Файл секретов CHAP

Он требует, чтобы вы имели взаимно опознавательные методы - вы должны позволить и вашей машине опознать удаленный сервер И удаленному серверу опознать вашу машину.

Итак, если ваша машина - fred и удаленная - barney, то ваша машина установила бы name fred remotename barney, а удаленная машина установит name barney remotename fred в их соответствующих файлах /etc/ppp/options.ttySx .

/etc/chap-secrets файл для fred такой

```
<p>
# Secrets for authentication using CHAP
# client      server  secret          acceptable local IP addresses
fred          barney  flintstone
barney        fred    wilma
```

а для barney

```
# Secrets for authentication using CHAP
# client      server  secret          acceptable local IP addresses
barney        fred    flintstone
fred          barney  wilma
```

Обратите внимание в частности, что обе машины должны иметь записи для двунаправленного установления подлинности. Это позволяет локальной машине называть себя на удаленной машине И удаленной машине называть себя на локальной машине.

13.5 Обработка нескольких PAP соединений

Некоторые пользователи работают с больше чем одним сервером, использующих PAP. Если ваше имя пользователя отличается для каждой машины, с которой вы хотите соединиться, это - не проблема.

Однако, многие пользователи имеют одно и то же имя пользователя на двух (или даже всех) системах, с которыми они соединяются. Вот это проблема - правильно выбрать соответствующие строки из файла `/etc/ppp/ppp-secrets`.

Как вы могли ожидать, PPP обеспечивает механизм для решения этой задачи. PPP позволяет вам установить 'вымышленное имя' для удаленного (серверного) конца соединения, используя опцию `remotename` в `pppd`.

Позвольте нам предположить, что вы соединяетесь с двумя PPP серверами, используя `username fred`. Вы настраиваете ваш `/etc/ppp/ppp-secrets` примерно так

```
fred    pppserver1    barney
fred    pppserver2    wilma
```

Теперь, чтобы установить соединение с `pppserver1`, вы использовали бы `name fred remotename pppserver1` в ваших `ppp`-опциях, а для `pppserver2` - `name fred remotename pppserver2`.

Поскольку вы можете выбирать файл опций для запуска `pppd`, используя опцию `file filename`, то вы можете настроить скрипт, чтобы соединяться с каждым из ваших PPP серверов, правильно выбирая файл опций для использования и, следовательно, выбирая правильную опцию `remotename`.

14 Установка PPP соединения вручную

Теперь, когда вы создали ваши файлы `/etc/ppp/options` и `/etc/resolv.conf` (и, в случае необходимости, файл `/etc/ppp/ppp|chap-secrets`), вы можете проверить установки, устанавливая PPP соединение вручную. (Как только мы настроим соединение вручную, то мы сможем автоматизировать процесс).

Чтобы сделать это, ваша коммуникационная программа должна уметь завершать свою работу БЕЗ сброса модема. `Minicom` может делать это - ALT Q (или в старых версиях `minicom` CTRL Q)

Удостоверьтесь, что вы зарегистрировались как `root`.

Запустите вашу коммуникационную программу (типа `minicom`), позвоните на PPP сервер и зарегистрируйтесь как обычно. Если вы должны выдать команду, чтобы запустить PPP на сервере, введите ее. Теперь вы увидите мусор, который вы видели прежде.

Если вы используете `ppp` или `chap`, то вы увидите мусор сразу после соединения с удаленной системой, без приглашений к регистрации (хотя это бывает не на всех серверах, попробуйте нажать `enter` и посмотрите, не появится ли мусор).

Теперь выйдите из коммуникационной программы без сброса модема (ALT Q или CTL Q в `minicom`) и в командной строке Linux (от `root`) напечатайте

```
pppd -d -detach /dev/ttySx 38400 &
```

`-d` опция включает отладку - все подробности установки `ppp` соединения будут записываться в ваш системный лог - что очень пригодится, если у вас появятся проблемы.

Лампочки на вашем модеме должны теперь вспыхивать, поскольку PPP соединение устанавливается. Это будет продолжаться короткое время, пока PPP соединения не будет установлено.

В этой точке вы можете посмотреть интерфейс PPP, дав команду

```
ifconfig
```

Кроме устройств ethernet и loopback вы должны увидеть еще кое-что вроде:

```
ppp0      Link encap:Point-Point Protocol
          inet addr:10.144.153.104  P-t-P:10.144.153.51  Mask:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:552  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
```

Где

- inet addr:10.144.153.10 IP адрес вашего конца соединения.
- P-t-P:10.144.153.5 IP адрес СЕРВЕРА.

(Естественно, ifconfig будет сообщать не эти IP адреса, а те, которые используются вашим PPP сервером.)

Обратите внимание: ifconfig также сообщает вам, что связь UP (активна) и RUNNING (работает)! Если вы не видите никаких ppp устройств или кое-что вроде

```
ppp0      Link encap:Point-Point Protocol
          inet addr:0.0.0.0  P-t-P:0.0.0.0  Mask:0.0.0.0
          POINTOPOINT  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
```

Ваше PPP соединение не было создано...см. раздел по отладке ниже!

Также вы должны увидеть маршрут к удаленному хосту (и другим внешним хостам). Чтобы сделать это, дайте команду

```
route -n
```

Вы должны увидеть что-то вроде:

```
Kernel routing table
Destination      Gateway          Genmask          Flags MSS      Window Use  Iface
10.144.153.3     *                255.255.255.255  UH    1500    0        1   ppp0
127.0.0.0        *                255.0.0.0        U     3584    0        11  lo
10.0.0.0         *                255.0.0.0        U     1500    0        35  eth0
default          10.144.153.3    *                UG    1500    0        5   ppp0
```

Обратите внимание, что мы имеем ДВЕ записи, указывающие на наш интерфейс ppp.

Первая - ХОСТОВЫЙ маршрут (обозначенный флажком H) и это позволяет нам видеть хост, с которым мы соединены - но ничего больше.

Вторая - маршрут, заданный по умолчанию (установленный опцией pppd defaultroute. Это - маршрут, который говорит, чтобы наш Linux PC посылал любые пакеты, НЕ предназначенные для локальной сети(ей) ethernet - для которой мы имеем свои сетевые маршруты - на PPP сервер непосредственно. Далее PPP сервер отвечает за маршрутизацию наших пакетов в Интернет и маршрутизацию ответных пакетов обратно к нам.

Если вы не видите таблицу маршрутизации с двумя записями, значит что-то неправильно. В частности, если ваш syslog показывает сообщение, сообщающее вам, что pppd не заменяет существующий заданный по умолчанию маршрут, то вы имеете заданный по умолчанию маршрут, указывающий на ваш ethernet интерфейс - который ДОЛЖЕН быть заменен в соответствии с указанным сетевым маршрутом: ВЫ МОЖЕТЕ ИМЕТЬ ТОЛЬКО ОДИН ЗАДАННЫЙ ПО УМОЛЧАНИЮ МАРШРУТ!!!

Вы должны будете исследовать ваши файлы инициализации системы, чтобы выяснить, где устанавливается этот заданный по умолчанию маршрут (используется команда route add default...). Измените эту команду на что-нибудь типа route add net....

Теперь проверьте связь 'пингуя' сервер по IP адресу как сообщено в результате выполнения команды ifconfig, то есть.

```
ping 10.144.153.51
```

Вы должны получить что-то вроде

```
PING 10.144.153.51 (10.144.153.51): 56 data bytes
64 bytes from 10.144.153.51: icmp_seq=0 ttl=255 time=328.3 ms
64 bytes from 10.144.153.51: icmp_seq=1 ttl=255 time=190.5 ms
64 bytes from 10.144.153.51: icmp_seq=2 ttl=255 time=187.5 ms
64 bytes from 10.144.153.51: icmp_seq=3 ttl=255 time=170.7 ms
```

Эта распечатка будет продолжаться до бесконечности - остановите ее нажатием CTRL C, и в этой точке вы получите более подробную информацию:

```
--- 10.144.153.51 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 170.7/219.2/328.3 ms
```

Итак, все хорошо.

Теперь попробуйте пинговать хост по имени (не имени PPP сервера непосредственно, а хоста в другом месте, про который вы ЗНАЕТЕ, что он наверняка работает). Например

```
ping sunsite.unc.edu
```

При этом появится небольшая пауза, поскольку Linux получает IP адрес по заданному FQDN, который вы 'пингуете' от DNS, который вы определили в файле /etc/resolv.conf - так что не волнуйтесь (но вы будете видеть мигание лампочек вашего модема). Короче, вы получите распечатку типа

```
PING sunsite.unc.edu (152.2.254.81): 56 data bytes
64 bytes from 152.2.254.81: icmp_seq=0 ttl=254 time=190.1 ms
64 bytes from 152.2.254.81: icmp_seq=1 ttl=254 time=180.6 ms
64 bytes from 152.2.254.81: icmp_seq=2 ttl=254 time=169.8 ms
64 bytes from 152.2.254.81: icmp_seq=3 ttl=254 time=170.6 ms
64 bytes from 152.2.254.81: icmp_seq=4 ttl=254 time=170.6 ms
```

Снова, остановите вывод, нажимая CTRL C, и получите статистику ...

```
--- sunsite.unc.edu ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 169.8/176.3/190.1 ms
```

Если вы не получаете ответа, попробуйте попинговать IP адрес DNS сервера вашего ISP. Если вы получаете результат от него, то похоже, что у вас проблема с файлом /etc/resolv.conf .

Если это не работает, то у вас проблема с маршрутизацией, или у вашего ISP проблема с маршрутизацией пакетов обратно к вам. Проверьте вашу таблицу маршрутизации как показано выше и если она в порядке, войдите в контакт с вашим ISP. Хороший тест ISP - использовать для соединения другую операционную систему. Если вы можете выбраться дальше вашего ISP, тогда проблема - на вашем конце.

Если все работает, закройте соединение, напечатав

```
ppp-off
```

После короткой паузы, модем должен повесить трубку.

Если это не работает, или выключите ваш модем или запустите вашу коммуникационную программу и прервите модем вводом + + +, и затем, получив ответ модема - ОК, "положите трубку" командой АТН0.

Возможно вам также понадобится очистить файл блокировки, созданный rpprd

```
rm -f /var/lock/LCK..ttySx
```

15 Автоматизация ваших соединений - Создание скриптов соединения

Хотя вы можете продолжать регистрироваться вручную как было показано выше, гораздо лучше настроить некоторые скрипты, чтобы для вас это происходило автоматически.

Набор скриптов автоматизирует процесс входа в систему и запускает PPP так что все, что вы должны сделать (от root или как член группы PPP) - дать одну команду запуска вашего соединения.

15.1 Скрипты соединения для аутентификации по имени/паролю пользователя

Если ваш ISP не требует использования PAP/CHAP, вам нужны именно эти скрипты!

Если пакет rppr установлен правильно, вы должны иметь два файла примеров. Для PPP 2.1.2 они находятся в /usr/sbin, а для PPP 2.2 они находятся в /etc/ppp/scripts. Они называются для PPP-2. 1.2

```
rppr-on rppr-off
```

```
а для PPP-2. 2
```

```
rppr-off rppr-on rppr-on-dialer
```

Теперь, если Вы используете PPP 2.1.2, я настойчиво прошу вас удалить файлы примеров. С ними имеются потенциальные проблемы - и не сообщайте мне, что они прекрасно работают - я использовал их очень долго (и даже рекомендовал их в первой версии этого HOWTO)!

Для пользователя PPP 2.1.2 имеется ЛУЧШАЯ версия шаблона, взятая из дистрибутива PPP 2.2 . Я предлагаю вам скопировать и использовать эти скрипты вместо старого скрипта PPP-2.1.2 .

15.2 Скрипт ppp-on

Это первый из ПАРЫ скриптов, которые фактически запускают соединение.

```
#!/bin/sh
#
# Скрипт для инициации соединения PPP. Это первая часть из пары скриптов.
# Это не секретные скрипты, так как коды видны командой ps.
# Однако это пример.
#
# Это параметры. Измените их как нужно.
TELEPHONE=555-1212      # Телефонный номер соединения
ACCOUNT=george         # Имя пользователя для входа ('George Burns')
PASSWORD=gracie        # Пароль для этого аккаунта (и 'Gracie Allen')
LOCAL_IP=0.0.0.0       # Локальный IP адрес, если известен. Динамический = 0.0.0.0
REMOTE_IP=0.0.0.0     # Удаленный IP адрес, если желателен. Обычно 0.0.0.0
NETMASK=255.255.255.0 # Соответствующая сетевая маска, если нужна
#
# Экспортируем их, чтобы они были доступны в 'ppp-on-dialer'
export TELEPHONE ACCOUNT PASSWORD
#
# Это расположение скрипта, который звонит по телефону и регистрируется в
# системе. Пожалуйста, используйте абсолютное имя файла, так как опция connect
# не использует переменную $PATH. (Если это сделать, то 'root' аккаунт будет
# дырой в защите, так что не просите.)
#
DIALER_SCRIPT=/etc/ppp/ppp-on-dialer
#
# Инициация соединения
#
#
exec /usr/sbin/pppd debug /dev/ttySx 38400 \
    $LOCAL_IP:$REMOTE_IP \
    connect $DIALER_SCRIPT
```

Это скрипт ppp-on-dialer:

```
#!/bin/sh
#
# Это вторая часть скрипта ppp-on. Она выполняет установку желаемого
# соединения.
#
/usr/sbin/chat -v
TIMEOUT          3
ABORT             '\nBUSY\r'
ABORT             '\nNO ANSWER\r'
ABORT             '\nRINGING\r\n\r\nRINGING\r'
''               \rAT
'OK-+++c-OK'     ATH0
TIMEOUT          30
OK               ATDT$TELEPHONE
CONNECT          ''
```

```

ogin:--ogin:      $ACCOUNT      \
assword:         $PASSWORD

```

Для PPP-2.2, сценарий rpp-off примерно такой:

```

#!/bin/sh
#####
#
# Определить прерываемое устройство.
#
if [ "$1" = "" ]; then
    DEVICE=ppp0
else
    DEVICE=$1
fi

#####
#
# Если pid файл ppp0 есть, тогда программа работает. Остановить ее.
if [ -r /var/run/$DEVICE.pid ]; then
    kill -INT `cat /var/run/$DEVICE.pid`
#
# Если kill не работает, тогда нет процесса, запущенного под этим pid.
# Это может также означать, что существует посторонний lock файл.
# Возможно, вы захотите удалить его.
    if [ ! "$?" = "0" ]; then
        rm -f /var/run/$DEVICE.pid
        echo "ERROR: Removed stale pid file"
        exit 1
    fi
#
# Отлично. Пусть rppd поправит свой собственный недочет.
    echo "PPP link to $DEVICE terminated."
    exit 0
fi
#
# для ppp0 не запущен ppp процесс
echo "ERROR: PPP link is not active on $DEVICE"
exit 1

```

15.3 Редактирование скриптов запуска PPP

Поскольку новые скрипты приходят двумя частями, мы в свою очередь отредактируем их.

15.3.1 Скрипт rpp-on

Вы должны будете отредактировать скрипт чтобы вставить ВАШЕ имя пользователя на вашем ISP, ВАШ пароль на вашем ISP, номер телефона вашего ISP.

Каждая из строк типа TELEPHONE = - это фактически установленные переменные shell, которые содержат информацию справа от = (исключая комментарии, конечно). Отредактируйте каждую из этих строк так, чтобы они соответствовали вашему ISP и соединению.

Также, поскольку вы устанавливаете IP адрес (если вам это нужно) в файле /etc/ppp/options, УДАЛИТЕ строку, которая говорит

```
$LOCAL_IP:$REMOTE_IP \
```

Также, удостоверьтесь, что переменная оболочки DIALER_SCRIPT указывает на полный путь и имя скрипта для дозвона, который вы фактически собираетесь использовать. Так что, если вы переместили его или переименовали скрипт, удостоверьтесь, что вы отредактировали эту строку правильно в скрипте ppp-on!

15.3.2 Сценарий ppp-on-dialer

Это - второй скрипт, который фактически поднимает нашу ppp связь.

Обратите внимание: chat скрипт обычно однострочный. Наклонные черты влево используются, чтобы разместить строки на нескольких физических строках (для удобочитаемости человеком) и не формировать часть скрипта самому.

Однако, очень полезно рассмотреть это подробно так, чтобы мы поняли, что же фактически (предположительно) происходит!

15.4 Что означает скрипт chat...

Скрипт chat - последовательность пар "ожидаемая строка/посылаемая строка". В частности обратите внимание, что мы ВСЕГДА ожидаем что-нибудь перед тем, как пошлем что-либо.

Если мы должны послать что-то БЕЗ того, чтобы сначала получить что-нибудь, мы должны использовать пустую строку ожидания (обозначаемую) и аналогично для ожидания чего-либо без того, чтобы послания чего-нибудь! Также, если строка состоит из нескольких слов, (например, NO CARRIER), вы должны взять строку в кавычки, чтобы chat воспринял ее как одно целое.

Строка chat в нашем шаблоне:

```
exec /usr/sbin/chat -v
```

В вызове chat опция -v говорит, чтобы chat копировал ВСЕ вводимое/выводимое в системный лог (обычно /var/log/messages). Как только вы убедитесь, что скрипт chat работает надежно, отредактируйте эту строку, чтобы удалить -v, чтобы не хранить ненужную информацию в вашем syslog.

```
TIMEOUT          3
```

Это устанавливает паузу для получения ожидаемого ввода в 3 секунды. Вы можете увеличить это значение до 5 или 10 секунд, если вы используете медленный модем!

```
ABORT            '\nBUSY\r'
```

Если получена строка BUSY, то операция аварийно прекращается.

```
ABORT          '\nNO ANSWER\r'
```

Если получена строка NO ANSWER, то операция аварийно прекращается.

```
ABORT          '\nRINGING\r\n\r\nRINGING\r'
```

Если (повторимся) получена строка RINGING, то операция аварийно прекращается. Это потому, что кто-то сидит на вашей телефонной линии!

```
"              \rAT
```

Не ожидаем ничего от модема, и послаем строку в него.

```
OK-+++ \c-OK   ATH0
```

Это немного более сложно, поскольку использует некоторые из возможностей восстановления при ошибках chat.

What is says is...Ожидаем ОК, если он не получен (потому что модем не в командном режиме), затем посылаем + + + (стандартная строка для Hayes-совместимых модемов, которая возвращает модем в командный режим) и ожидаем ОК.

Затем посылаем ATH0 (строка для завершения связи модема). Это позволяет вашему скрипту справиться с вашим модемом, зависшим во время сеанса связи!

```
TIMEOUT        30
```

Установим паузу по времени в 30 секунд для оставшихся команд скрипта. Если вы испытываете проблемы со скриптом chat, прерывающимся из-за пауз, увеличьте это значение до 45 секунд или больше.

```
OK              ATDT$TELEPHONE
```

Ожидаем ОК (ответ модема на команду ATH0) и набираем номер, на который мы хотим позвонить.

```
CONNECT        ''
```

Дожидаемся строки CONNECT (которую наш модем посылает, когда удаленные модем отвечает) и не посылаем в ответ ничего.

```
ogin:--ogin:      $ACCOUNT
```

Снова, здесь мы вставляем кое-какое исправление ошибок. Ожидаем подсказку входа в систему (...ogin:), но если мы не получаем ее по истечении паузы, то посылаем возврат каретки и затем ищем подсказку входа в систему снова. Когда подсказка получена, посылаем username (сохраненное в переменной shell \$ACCOUNT).

```
assword:          $PASSWORD
```

Ожидаем запроса пароля и посылаем наш пароль (аналогично сохраненный в переменной shell). Этот скрипт chat имеет приемлемую возможность исправления ошибок. chat имеет значительно большее количество возможностей, чем показано здесь. Для подробной информации проконсультируйтесь с man chat (man 8 chat).

15.4.1 Запуск PPP на серверной стороне моединения

Хотя скрипт rpp-on-dialer отлично подходит для серверов, которые автоматически запускают rppd на серверной стороне как только вы зарегистрировались, некоторые серверы требуют, чтобы вы явно дали команду запуска PPP на сервере.

Если вы должны дать команду, чтобы запустить PPP на сервере, вы должны отредактировать скрипт rpp-on-dialer.

В КОНЦЕ сценария (после строки пароля) добавьте дополнительную пару ожидаемой-посылаемой строк - которая искала бы вашу подсказку регистрации в системе (отличая символы, которые имеют специальное значение в оболочке Bourne - типа \$ и [или] (открытые и закрытые квадратные скобки)).

Как только программа chat нашла командную строку оболочки, она должна выдать команду запуска rpp, требуемую для PPP сервера вашего ISP.

В моем случае, мой PPP сервер использует стандартную подсказку bash Linux.

```
[hartr@kepler hartr]$
```

и требует, чтобы я напечатал

```
rpp
```

чтобы запустить PPP на сервере.

Хорошо бы учесть здесь некоторые ошибки, так например в моем случае я использую

```
hartr--hartr      rpp
```

Это значит, что если мы не получили подсказку в течении заданной паузы, то посылаем возврат каретки и ищем подсказку снова.

Как только подсказка получена, посылаем строку ppp.

Обратите внимание: не забудьте добавить \ к концу предыдущей строки, так что бы chat думал, что весь сценарий chat состоит из одной строки!

К сожалению, на некоторых серверах набор подсказок часто меняется!

Возможно вам понадобится несколько раз зарегистрироваться, используя minicom, чтобы понять, что происходит, и найти устойчивую "ожидаемую" строку.

15.5 Сценарий chat для соединения PAP/CHAP

Если ваш ISP использует PAP/CHAP, то ваш сценарий chat намного более простой.

Весь ваш сценарий chat должен делать вот что - звонить по номеру телефона, ждать соединения и затем позволить pppd обработать регистрацию в системе!

```
#!/bin/sh
#
# This is part 2 of the ppp-on script. It will perform the connection
# protocol for the desired connection.
#
exec /usr/sbin/chat -v \
    TIMEOUT          3 \
    ABORT             '\nBUSY\r' \
    ABORT             '\nNO ANSWER\r' \
    ABORT             '\nRINGING\r\n\r\nRINGING\r' \
    ''               \rAT \
    'OK-+++\c-OK'    ATH0 \
    TIMEOUT          30 \
    OK                ATDT$TELEPHONE \
    CONNECT          '' \
```

15.6 Отладка pppd и опция file option_file

Как мы уже видели, вы можете включить отладочную информацию опцией -d в pppd. Опция 'debug' эквивалентна ей.

Поскольку мы устанавливаем новое соединение с новым скриптом, сейчас поставьте опцию отладки. (Предупреждение: если у вас мало дискового пространства, то логи pppd могут быстро увеличить ваш файл syslog и создать вам проблему - but to do this you must fail to connect and keep on trying for quite a few minutes).

Как только вы обрадуетесь, что все работает правильно, можете удалить эту опцию.

Если вы назвали ваш ppp файл опций как-нибудь иначе, чем /etc/ppp/options, или /etc/ppp/options.ttySx, определите имя файла опцией file в pppd - например

```
exec /usr/sbin/pppd debug file options.myserver /dev/ttyS0 38400 \
```

16 Тестирование вашего сценария соединения

Откройте новый root Xterm (если вы находитесь в X) или откройте новую виртуальную консоль и войдите в систему как пользователь root.

В этом новом сеансе, выдайте команду

```
tail -f /var/log/messages
```

(или какой-то другой файл вашего системного лога).

В первом окне (или виртуальной консоли) выдайте команду

```
ppp-on &
```

(или как вы там назвали вашу отредактированную версию /usr/sbin/ppp-on). Если вы не перевели запущенный скрипт в фоновый режим, поместив в конце команды, &, то вы не получите назад приглашение командной строки до тех пор, пока ppp не закончит работу (пока связь не завершится).

Теперь переключитесь обратно на окно, которое отслеживает ваш системный файл регистрации. Вы увидите что-то вроде следующего (при условии что вы указали -v для chat и -d для pppd) это - сценарий chat и ответы, регистрируемые в системном логе, и следующей далее информацией от pppd:

```
Oct 21 16:09:58 hwin chat[19868]: abort on (NO CARRIER)
Oct 21 16:09:59 hwin chat[19868]: abort on (BUSY)
Oct 21 16:09:59 hwin chat[19868]: send (ATZ^M)
Oct 21 16:09:59 hwin chat[19868]: expect (OK)
Oct 21 16:10:00 hwin chat[19868]: ATZ^M^M
Oct 21 16:10:00 hwin chat[19868]: OK -- got it
Oct 21 16:10:00 hwin chat[19868]: send (ATDT722298^M)
Oct 21 16:10:00 hwin chat[19868]: expect (CONNECT)
Oct 21 16:10:00 hwin chat[19868]: ^M
Oct 21 16:10:22 hwin chat[19868]: ATDT722298^M^M
Oct 21 16:10:22 hwin chat[19868]: CONNECT -- got it
Oct 21 16:10:22 hwin chat[19868]: send (^M)
Oct 21 16:10:22 hwin chat[19868]: expect (ogin:)
Oct 21 16:10:23 hwin chat[19868]: kepler login: -- got it
Oct 21 16:10:23 hwin chat[19868]: send (hartr^M)
Oct 21 16:10:23 hwin chat[19868]: expect (ssword:)
Oct 21 16:10:23 hwin chat[19868]: hartr^M
Oct 21 16:10:23 hwin chat[19868]: Password: -- got it
Oct 21 16:10:23 hwin chat[19868]: send (??????^M)
Oct 21 16:10:23 hwin chat[19868]: expect (hartr)
Oct 21 16:10:24 hwin chat[19868]: [hartr -- got it
Oct 21 16:10:24 hwin chat[19868]: send (ppp^M)
Oct 21 16:10:27 hwin pppd[19872]: pppd 2.1.2 started by root, uid 0
Oct 21 16:10:27 hwin pppd[19873]: Using interface ppp0
Oct 21 16:10:27 hwin pppd[19873]: Connect: ppp0 <--> /dev/cua1
Oct 21 16:10:27 hwin pppd[19873]: fsm_sdata(LCP): Sent code 1, id 1.
Oct 21 16:10:27 hwin pppd[19873]: LCP: sending Configure-Request, id 1
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconfreq(LCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd MRU
Oct 21 16:10:27 hwin pppd[19873]: (1500)
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd ASYNCMAP
Oct 21 16:10:27 hwin pppd[19873]: (0)
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd MAGICNUMBER
Oct 21 16:10:27 hwin pppd[19873]: (a098b898)
```

```
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd PCOMPRESSION
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd ACCOMPRESSION
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: returning CONFACK.
Oct 21 16:10:27 hwin pppd[19873]: fsm_sdata(LCP): Sent code 2, id 1.
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconfack(LCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 1, id 1.
Oct 21 16:10:27 hwin pppd[19873]: IPCP: sending Configure-Request, id 1
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconfreq(IPCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: ipcp: received ADDR
Oct 21 16:10:27 hwin pppd[19873]: (10.144.153.51)
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: ipcp: received COMPRESSTYPE
Oct 21 16:10:27 hwin pppd[19873]: (45)
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: ipcp: returning Configure-ACK
Oct 21 16:10:28 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 2, id 1.
Oct 21 16:10:30 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 1, id 1.
Oct 21 16:10:30 hwin pppd[19873]: IPCP: sending Configure-Request, id 1
Oct 21 16:10:30 hwin pppd[19873]: fsm_rconfreq(IPCP): Rcvd id 255.
Oct 21 16:10:31 hwin pppd[19873]: ipcp: received ADDR
Oct 21 16:10:31 hwin pppd[19873]: (10.144.153.51)
Oct 21 16:10:31 hwin pppd[19873]: (ACK)
Oct 21 16:10:31 hwin pppd[19873]: ipcp: received COMPRESSTYPE
Oct 21 16:10:31 hwin pppd[19873]: (45)
Oct 21 16:10:31 hwin pppd[19873]: (ACK)
Oct 21 16:10:31 hwin pppd[19873]: ipcp: returning Configure-ACK
Oct 21 16:10:31 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 2, id 255.
Oct 21 16:10:31 hwin pppd[19873]: fsm_rconfack(IPCP): Rcvd id 1.
Oct 21 16:10:31 hwin pppd[19873]: ipcp: up
Oct 21 16:10:31 hwin pppd[19873]: local IP address 10.144.153.104
Oct 21 16:10:31 hwin pppd[19873]: remote IP address 10.144.153.51
```

(Примечание - я использую СТАТИЧЕСКИЕ IP адреса - следовательно моя машина, отправила это PPP серверу - вы не увидите это, если используете ДИНАМИЧЕСКИЕ IP адреса.) Также, этот сервер требует дать команду для запуска ppp.

Как будто все в порядке - протестируем соединение как и прежде, пингуя IP адреса и имена машин. Запустите ваш веб браузер или еще что-нибудь и скользите по сети - вы подключены!

17 Заккрытие PPP связи

Для завершения связи PPP используйте стандартную команду ppp-off, чтобы закрыть его (помните - вы должны быть root'ом или членом группы PPP!).

В вашем системном логе вы увидите что-то вроде:

```
Oct 21 16:10:45 hwin pppd[19873]: Interrupt received: terminating link
Oct 21 16:10:45 hwin pppd[19873]: ipcp: down
Oct 21 16:10:45 hwin pppd[19873]: default route ioctl(SIOCDELRT): Bad address
Oct 21 16:10:45 hwin pppd[19873]: fsm_sdata(LCP): Sent code 5, id 2.
Oct 21 16:10:46 hwin pppd[19873]: fsm_rtermack(LCP).
```

```
Oct 21 16:10:46 hwin pppd[19873]: Connection terminated.  
Oct 21 16:10:46 hwin pppd[19873]: Exit.
```

Не волнуйтесь о SIOCDELRT - это только замечание rppd о том, что он завершает работу и незачем волноваться.

18 Отладка

Есть множество причин, по которым ваше соединение не будет работать, chat не сумел завершиться правильно, неправильная строчка и т.д. Проверьте записи в вашем syslog.

18.1 Я вкомпилировал поддержку PPP в ядро, но ...

Самая общая проблема состоит в том, что люди компилируют поддержку PPP в ядро и тем не менее, когда они пытаются выполнить rppd, ядро жалуется, что оно не поддерживает rppd! Имеется ряд причин, по этому это может происходить.

18.1.1 Хотя вы пересобрали ваше ядро с поддержкой rppd, вы не загрузили новое ядро. Это может случиться, если вы не модифицировали /etc/lilo.conf и не перезапустили lilo. Вы загружаете правильное ядро?

Хорошая проверка для ядра - команда uname -a, которая должна вывести

```
Linux archenland 2.0.28 #2 Thu Feb 13 12:31:37 EST 1997 i586
```

Выведены версия ядра и дата, когда это ядро компилировалось - этого должно быть достаточно, чтобы разобраться с данным вопросом.

18.1.2 Вы вкомпилировали поддержку rppd в ядре как модуль?

Вы можете получить такую ошибку, если вы вкомпилировали поддержку rppd в ваше ядро как модуль, но сам модуль не построили и не установили. Посмотрите Kernel-HOWTO и файл README в /usr/src/linux!

Другая версия, относящаяся к проблеме с модулем, состоит в том, что вы ожидаете, что требуемые модули будут автоматически загружены, но не запустили демон kerneld (который на лету автоматически загружает и выгружает модули).

Проверьте kerneld mini-HOWTO для информации по настройке kerneld.

18.1.3 Вы используете правильную версию PPP для вашего ядра?

Вы должны использовать rppd-2.2 с ядром версии 2.0.x. Вы можете использовать rppd-2.2 с ядром версии 1.2.x (если вы отпатчили ядро) иначе вы должны использовать rppd-2.1.2.

18.1.4 Вы запускаете rppd от root?

Если вы не запускаете rppd от пользователя root (и rppd - не suid для root), то вы может получить это сообщение.

18.2 Мой модем соединяется, но ppp не запускается

На это может иметься бесконечное множество причин (см. comp.os.linux...).

САМАЯ основная ошибка - вы имеете опечатки в ваших скриптах. Единственное, что здесь можно сделать - это удостовериться, в правильной работе сценария chat, пролистав ваш syslog (/var/log/messages) строчка за строчкой.

Можно попробовать попытаться соединиться с PPP сервером вручную, чтобы проверить не изменились ли условия регистрации в системе.

Вы должны очень тщательно проверить лог-файл и посмотреть там реально выдаваемые подсказки и имейте в виду, что мы - люди часто заменяем в своем воображении фактически написанный текст на тот, который нам показался написанным на этом месте!

18.3 Syslog говорит "serial line is not 8 bit clean..."

в этом случае также возможны варианты - например последовательная линия looped обратно и т.д., и происходить это может по ряду причин.

Чтобы понять, что происходит, необходимо немного углубиться в процессы, происходящие в rppd непосредственно.

Когда rppd запускается, он посылает LCP (протокол управления связи) пакеты удаленной машине. Если он получает приемлемый ответ, то переходит в следующую стадию (используя IPCP пакеты) и только когда эти переговоры завершаются - начинает действовать IP уровень так, чтобы вы могли использовать связь PPP.

Если на удаленном конце линии нет ppp сервера, то когда ваш PC посылает lcp пакеты, они возвращаются искаженными процессом login на удаленной конце. Поскольку эти пакеты используют 8 битов, а вернувшиеся пакеты приходят с отрезанным 8-ым битом (вспомните, что ASCII - 7 разрядный код), то PPP видит это и соответственно жалуется.

Имеется несколько причин такого отражения сигналов.

18.3.1 Вы неправильно регистрируетесь на сервере

Когда ваш сценарий chat завершается, на вашем PC запускается rppd. Однако, если вы не завершили процесс входа в систему на сервере (включая посылку команды, требуемой для запуска PPP на сервере), то PPP не запустится.

Так lcp пакеты отражаются, и вы получаете эту ошибку.

Вы должны тщательно проверить и исправить (в случае необходимости) ваш сценарий chat (см. выше).

18.3.2 Вы не запустили PPP на сервере

Некоторые PPP серверы требуют, чтобы вы ввели команду и/или нажали return после завершения процесса регистрации, чтобы на удаленном конце стартовал ppp.

Проверьте ваш сценарий chat (см. выше).

Если вы регистрируетесь вручную и обнаруживаете, что после этого вы должны послать return, чтобы запустить PPP, то просто добавьте пустую пару "ожидаемое- посылаемое" в конец вашего сценария chat (пустая "посылаемая" строка фактически посылает return).

18.3.3 Удаленный PPP процесс медленно запускается

This one is a bit tricky!

По умолчанию, ваш Linux rppd скомпилирован для посылки максимум 10 lcp запросов конфигурации. Если сервер медленно отвечает, то все 10 таких запросов могут передаться до того, как удаленный PPP будет готов получить их.

На вашей машине rppd видит, что все 10 запросов отражены обратно (с 8-ым отрезанным битом) и завершается.

Имеются два способа обхода:

Добавьте `lcp-max-configure 30` в ваши опции `ppp`. Таким образом увеличивается максимальное число посылаемых `lcp` пакетов выбора конфигурации. Для действительно медленных серверов вам может понадобиться указать еще большее количество таких пакетов.

В качестве альтернативы вы можете `get a bit tricky in return`. Вы, возможно, заметили, что, когда вы регистрировались вручную на PPP сервере, и PPP там запускался, то первый символ `ppp` мусора был всегда символ тильды (`~`).

Это наблюдение можно использовать таким образом - мы можем добавить новую пару "ожидаемое-посылаемое" в конец вашего сценария `chat`, которая будет ожидать тильду и не посылать ничего. Это можно сделать, например, так:

```
\~      ''
```

Обратите внимание: поскольку символ тильды имеет специальное значение в `shell`, то нужно его за'`escape`'ить (и, следовательно, перед ним поставить наклонную черту влево).

18.4 Заданный по умолчанию маршрут не установлен

Если `pppd` отказывается установить маршрут заданный по умолчанию, то, поэтому он (совершенно правильно) отказывается удалять/заменять существующий заданный по умолчанию маршрут.

Обычная причина состоит в том, что некоторые дистрибутивы заданный по умолчанию маршрут устанавливают через `ethernet-адаптер`, а не через сеть.

См. `NAG Linux` и `Net2/3 HOWTOS` для информации о правильной установке вашей платы `ethernet` и связанных с нею маршрутов.

Также возможно, что ваша LAN уже использует шлюз/маршрутизатор и в вашей таблице маршрутизации на него был установлен маршрут по умолчанию.

Исправление этой последней ситуации может требовать некоторых знаний IP сетей и лежит не охватывается этим `HOWTO`. Предлагаю вам обратиться к экспертам (в группы новостей).

18.5 Другие проблемы

Имеется много причин, кроме перечисленных, по которым `ppp` будет не в состоянии соединиться и/или функционировать правильно.

Смотрите `PPP FAQ` (который является набором вопросов и ответов). Это - очень всесторонний документ, и ответы там ЕСТЬ! Из моего собственного (грустного) опыта, если там нет ответа на ваши вопросы, то проблема - НЕ в `ppp`! В моем случае я использовал ELF ядро, которое я не нарастил до возможностей, соответствующих ядерным модулям.

Я потратил впусую около 2 дней (и часть ночи), ругая прекрасно работающий PPP сервер, пока меня не осенило!

19 Получение помощи, когда не знаете, что еще можно сделать

Если вы не можете настроить связь PPP, вернитесь в начало документа и проверьте все - в особенности информацию, выводимую `"chat -v ..."` и `"pppd -d"` в вашем системном логе.

Также проконсультируйтесь с документацией по PPP и FAQ плюс другими упоминаемыми здесь документами!

Если и это не помогает, попробуйте попросить помощи у людей в группах новостей `comp.os.linux.misc` и `comp.os.linux.networking` или в `comp.protocols.ppp`

Вы можете попробовать послать мне персональную почту, но я у меня есть своя работа (и личная жизнь), и я не гарантирую, что ответу быстро (если вообще смогу ответить), поскольку это зависит от моей текущей загрузки и состояния моей личной жизни!

В частности - НЕ НАДО ОТПРАВЛЯТЬ КУЧУ ОТЛАДОЧНОЙ ИНФОРМАЦИИ НИ В ГРУППЫ НОВОСТЕЙ, НИ МНЕ НА EMAIL - все это будет отправлено в /dev/null (если я не просил специально сделать это).

20 Общие проблемы, появляющиеся, когда связь заработает

Одна проблема, которую вы обнаружите, состоит в том, что многие поставщики услуг будут поддерживать только тот коммуникационный программный пакет, который они распространяют для новых учетных записей. Это - (обычно) Microsoft Windows :- (- и многие службы помощи ISP, кажется, ничего не могут посоветовать относительно UNIX (или Linux). Так, что будьте готовы к тому, что помощи от них вы не получите!

You could of course do the individual a favour and educate them about Linux (any ISP help desk person should be reasonably 'with it' in Internet terms and that means they should have a home Linux box - of course it does)!

20.1 Я не могу выйти дальше PPP сервера, к которому я подсоединяюсь

Хорошо - ваше PPP соединение установлено, и вы можете пинговать IP адрес PPP сервера (второй или "удаленный" IP адрес показывается командой `ifconfig ppp0`), но вы не можете выйти дальше этого PPP сервера.

Прежде всего, попробуйте пинговать IP адреса, которые вы определили в `/etc/resolv.conf` как DNS. Если это работает, то вы можете выйти дальше вашего PPP сервера (если DNS имеет не тот же самый IP адрес, как и "удаленный" IP адрес вашего соединения). Теперь попробуйте пинговать полное Internet имя вашего поставщика услуг, например

```
ping my.provider.net.au
```

Если это не работает, то у вас проблему с преобразованием доменных имен.

Вероятно это из-за опечатки в вашем файле `/etc/resolv.conf`.

Тщательно проверьте всю информацию, касающуюся вашего поставщика услуг. Если все правильно, позвоните вашему поставщику услуг и убедитесь, что вы правильно записали IP адреса. Если соединение ВСЕ ЕЩЕ не работает (и ваш поставщик услуг подтверждает, что его DNS работают правильно), то у вас проблемы где-то еще, и я предлагаю вам тщательно проверить вашу установку Linux (особенно рекомендую проверить прав доступа к файлам).

Если вы ВСЕ ЕЩЕ не можете пинговать IP DNS вашего поставщика услуг по IP адресам, или они в действительности не работают (сделайте голосовой звонок и проверьте) или это проблема маршрутизации у вашего поставщика услуг. Снова, позвоните им и предложите проверить это.

Одна из версий заключается в том, что на "удаленном конце" стоит Linux PPP сервер, на котором опция IP форвардинга не была определена в ядре!

Хороший общий тест - это попробовать соединиться с вашим поставщиком услуг, используя программное обеспечение, которым пользуется большинство, для Microsoft Windows. Если все работает на другой операционной системе для этого же аккаунта, то проблема - с вашей Linux системой, а НЕ вашим поставщиком услуг.

20.2 Я могу посылать email, но не могу получить его

Если вы используете динамические IP адреса, то это совершенно нормально. См. "Настройка услуг" ниже.

20.3 Почему на моей машине не работают finger, WWW, gopher, talk итд?

Снова, если вы используете динамические IP адреса, это совершенно нормально. См. "Настройка Услуг" ниже.

21 Использование услуг Интернет с динамическими IP адресами

Если вы используете динамические IP адреса (и многие поставщики услуг дадут вам только динамический IP адрес, если вы не захотите платить значительно больше для вашего соединения), то вы должны знать об ограничениях, которые при этом налагаются.

Прежде всего, прекрасно будут работать только исходящие запросы на обслуживание. То есть вы можете посылать email используя sendmail (если он у вас правильно установлен), скачивать по ftp файлы из удаленных мест, использовать finger на другие машины, бродить по веб и т.д.

В частности, вы можете отвечать на email, который вы сбросили на вашу машину, хотя вы отключены от линии. Почта будет просто находиться в вашей почтовой очереди, пока вы не позвоните опять вашему ISP.

Однако, ваша машина не соединена с Интернетом 24 часа в день и вдобавок не имеет один и тот же IP адрес при каждом соединении. Так что вы не можете получить email, отправленный вашей машине, и очень трудно установить такой веб или ftp сервер, к которому могут обращаться ваши друзья! Относительно Интернет, ваша машина не существует как уникальная, постоянно доступная машина, поскольку не имеет уникального IP адреса (помните - другие машины будут использовать этот IP адрес, когда они звонят вашему провайдеру).

Если вы устанавливаете WWW (или любой другой сервер), это полностью неизвестно остальным пользователям Интернет, ЕСЛИ они не знают, что ваша машина сейчас соединена с Интернет И каков ваш текущий IP адрес. Имеется ряд способов, которым они могут получать эту информацию, надо им об этом сообщить либо почтой, либо хитрым использованием ".plan" файлов в shell аккаунта вашего поставщика услуг (полагая, что ваш поставщик позволяет доступ к shell и finger).

Теперь, для большинства пользователей, это не проблема - все, что хочет большинство людей - это посылать и получать email (используя учетную запись поставщика услуг) и бродить по WWW, ftp и другим серверам в Интернет. Если вы ДОЛЖНЫ осуществлять входящие соединения с вашим сервером, вы должны получить статический IP адрес. В качестве альтернативы вы можете исследовать методы, упомянутые выше ...

21.1 Установка email

Даже для динамических IP чисел, вы, конечно, можете настроить sendmail на вашей машине для отправки любого email, который вы написали локально.

Конфигурация sendmail может быть трудной - так что этот документ не пытается сообщать вам, как это сделать. Однако, вы вероятно должны сконфигурировать sendmail так, чтобы ваш поставщик услуг Интернет был обозначен как ваш "smart host"(sendmail.cf - опция DS). (Для подробностей по настройке sendmail см. документацию sendmail - и посмотрите конфигурации m4, которые поставляются с sendmail. Там имеется почти все, что может вам потребоваться).

Имеются также превосходные книги по Sendmail (особенно 'библия' от O'Reilly и Ассоциации), но эти книги - почти массовое убийство для большинства пользователей!

Как только вы сконфигурировали sendmail, вы вероятно захотите чтобы sendmail посылал любые сообщения, которые стоят в исходящей почтовой очереди, при установлении PPP соединения. Чтобы сделать это, добавьте команду

```
sendmail -q &
```

в ваш скрипт /etc/ppp/ip-up (см. ниже).

Входящая почта - это проблема для динамических IP адресов. Способ обрабатывать ее:

настройте вашего почтового агента пользователя так, чтобы вся почта была выпущена с заголовком "reply to", дающим ваш email адрес, выданный вашим поставщиком услуг Интернета.

Если вы можете, вы должны также установить ваш FROM адрес, чтобы ваш адрес email, выданный вашим ISP был правилен.

Используйте программы popclient, fetchmail, чтобы забрать вашу почту с вашего поставщика услуг. В качестве альтернативы, если ваш ISP использует IMAP, используйте почтового агента пользователя, умеющего работать с IMAP (типа pine).

Вы можете автоматизировать этот процесс в случае модемного соединения, помещая необходимые команды в скрипте `/etc/ppp/ip-up` (см. ниже).

21.2 Установка локального сервера доменных имен

Хотя вы можете совершенно спокойно использовать сервер доменных имен вашего ISP, также вы можете установить локальный кэширующий (вторичный) сервер имен, который указывается в скрипте `ip-up`. Преимущество локального (кэширующего) сервера доменных имен в том, что он сэкономит вам время (и снизит трафик), если вы часто заходите на одни и те же места Интернет в течение долгого интерактивного сеанса.

Настройка DNS в качестве кэширующего сервера имен (который использует строку "forwarders" в файле `named.boot`, указывающем на DNS вашего ISP) относительно проста. Книга O'Reilly (DNS и Bind) объясняет все, что вы хотите знать об этом.

Имеется также DNS-HOWTO.

Если вы работаете с маленькой LAN, которая может обращаться к Интернету через ваш Linux PC (используя IP Masquerade, например), то вероятно хорошая идея запустить локальный сервер имен (с директивой `forwarders`) пока связь установлена поскольку это минимизирует трафик и задержки, связанные с определением IP адреса по доменному имени.

Тонкость сетевого этикета: спросите разрешение вашего ISP, прежде чем вы начнете использовать вторичный кэширующий сервер имен в домене вашего ISP. Правильно сконфигурированный ваш DNS не будет вызывать никаких проблем для вашего ISP вообще, но если вы настроите что-то неправильно, то это может привести к проблемам.

22 Связывание двух сетей, использующих PPP

В основном нет различия между соединением одного Linux PC с PPP сервером и соединением двух LAN, использующих PPP на машине в каждой LAN.

Помните, PPP - это протокол точка-точка.

Однако, вы ОПРЕДЕЛЕННО должны понимать как устанавливается маршрутизация. Читайте NET-2 howto и Руководство Сетевого Администратора Linux (NAG). Вы также найдете неоценимую помощь в книге "Сетевое администрирование TCP/IP" (изданной O'Reilly и Assoc - ISBN 0-937175-82-X).

Если вы собираетесь работать с подсетями на обеих сторонах связи, то вам также надо найти Linux subnetworking mini-howto). Он доступен в *Linux Subnetworking mini-HOWTO*.

Для того, чтобы связать две LAN, вы должны использовать различные сетевые IP адреса (или подсети одного и того же сетевого адреса) и должны будете использовать статические IP адреса - или использовать IP masquerade. Если вы хотите использовать IP masquerade, см. IP masquerade mini-howto для инструкций по его установке.

22.1 Установка IP адресов

Договоритесь с сетевым администратором другой LAN о IP адресах, которые будут использоваться на каждом конце интерфейса PPP. Если вы используете статические IP адреса, то также вероятно потребуются, чтобы вы звонили по определенному номеру телефона.

Теперь отредактируйте соответствующий файл `/etc/ppp/options[.ttyXX]` - это хорошая идея - иметь специальный модем и порт на вашем конце для этого соединения. Может потребоваться изменить ваш файл `/etc/ppp/options` - и создать соответствующие файлы `options.ttyXX` для любых других соединений!

Определите IP адреса на вашем конце PPP соединения в соответствующем файле опций точно так, как указано выше для статических IP адресов.

22.2 Установка маршрутизации

Вы должны упорядочить прохождение пакетов вашей LAN по интерфейсу, который устанавливает PPP связь. Это - двустадийный процесс.

Прежде всего, вы должны установить маршрут от машины, выполняющей PPP соединение с сетью(ями) на удаленном конце связи. Если связь с Интернет, то она может быть обработана в соответствии с маршрутом заданным по умолчанию, установленным непосредственно rppd на вашем конце соединения, используя опцию 'defaultroute' rppd.

Если однако, связь связывает только две LAN, то специальный сетевой маршрут добавляется для каждой сети, которая доступна через соединение. Это делается с использованием команды 'route' для каждой сети в скрипте /etc/ppp/ip-up (см. После того, как связь установлена... для инструкций о том, как это выполнить).

Вторая вещь, которую вы должны сделать, это сообщить другим компьютерам о вашей LAN, что ваш Linux компьютер является фактически 'шлюзом' для сети(ей) на удаленный конец rppr соединения. Конечно, сетевой администратор на другой стороне соединения также должен проделать все это! Однако, так как он/она будет направлять пакеты в указанные вами сети, то потребуются специфический сетевой маршрут, не заданный по умолчанию (unless the LANs at the far end of the link are linking into you to access the Internet across your connection).

22.3 Сетевая безопасность

Если Вы связываете вашу LAN с Интернетом, используя PPP - или даже только к "чужой" LAN, вы должны подумать о проблеме защиты. Я настойчиво прошу вас подумать об установке firewall! Вы должны также поговорить об этом с администратором LAN вашего сайта ПЕРЕЖДЕ, ЧЕМ вы начнете связываться с посторонними LAN или Интернет. Неправильная настройка может привести к неизвестно чему - от отсутствия какой-либо реакции до действительно серьезной проблемы!

23 После того, как связь установлена - скрипт /etc/ppp/ip-up

Как только связь PPP установлена, rppd ищет /etc/ppp/ip-up. Если этот сценарий существует и может выполняться, PPP демон выполняет скрипт. Он позволяет вам автоматизировать любые специальные команды маршрутизации, которые могут вам понадобиться, и любые другие действия, которые вы хотите выполнить при активизации PPP соединения.

Это только скрипт shell, и он может делать все, что может делать скрипт shell (то есть фактически все что вы хотите).

Например, вы можете заставить sendmail послать исходящую почту, стоящую в очереди.

Точно так же вы можете вставить команды в ip-up для забора почты (используя POP), ждущий вас на вашем ISP.

Имеются ограничения на /etc/ppp/ip-up:

Он выполняется в преднамеренно ограниченном окружении, чтобы усилить защиту. Это означает, что вы должны указывать полный путь к запускаемым файлам и т.д.

Технически /etc/ppp/ip-up - это программа, а не скрипт. Это означает, что он может быть непосредственно выполнен - и следовательно требуется стандартный магический файл (#!/bin/bash) в начале первой строки скрипта и он должен быть читаем и выполнен пользователем root.

23.1 Специальная маршрутизация

Если вы связываете две LAN, вы будете должны установить специфические маршруты к 'посторонним' LAN. Это легко выполняется, используя скрипт /etc/ppp/ip-up. Единственная трудность возникает, если ваша машина работает с несколькими PPP соединениями.

Это потому, что /etc/ppp/ip-up выполняется для КАЖДОГО rppr соединения, которое устанавливается, так что вы должны тщательно выполнить правильные команды маршрутизации

для определенного устанавливаемого соединения - и не выполнять их, когда устанавливается любое другое соединение!

23.2 Обработка почтовой очереди

Когда появляется связь между двумя LAN, вы можете захотеть удостовериться, что почта, которая поставлена в очередь на обоих концах соединения, отправлена адресату. Это выполняется добавлением соответствующего вызова sendmail.

Для этого используется выражения bash 'case' для соответствующего параметра, передаваемого скрипту, запускающему rppd. Например вот скрипт /etc/ppp/ip-up, который я использую, чтобы обработать наши WAN соединения и связь с моей LAN ethernet (также обрабатываемую на том же самом rpp сервере).

23.3 Пример скрипта /etc/ppp/ip-up

Пример ниже обеспечивает ряд примеров использования.

```
#!/bin/bash
#
# Скрипт, выполняющий при необходимости задачи маршрутизации для rppd
# Только связь с Newman требует такой обработки.
#
# Когда rpp связь установлена, этот скрипт вызывается со следующими
# параметрами
#   $1      имя интерфейса, используемое rppd (напр. rpp3)
#   $2      имя устройства tty
#   $3      скорость устройства tty
#   $4      локальный IP адрес для интерфейса
#   $5      удаленный IP адрес
#   $6      параметр, указанный опцией 'ipparam' в rppd
#
case "$5" in
# Обработка маршрутизации для сервера Newman Campus
  202.12.126.1)
    /sbin/route add -net 202.12.126.0 gw 202.12.126.1
# and flush the mail queue to get their email there asap!
    /usr/sbin/sendmail -q &
    ;;
  139.130.177.2)
# Наше Internet соединение
# когда связь будет установлена, запустить сервер времени и синхронизировать с
# миром. Предполагается, что он еще не запущен
    if [ ! -f /var/lock/subsys/xntpd ]; then
        /etc/rc.d/init.d/xntpd.init start &
    fi
# Запуск сервера новостей (если еще не запущен)
    if [ ! -f /var/lock/subsys/news ]; then
        /etc/rc.d/init.d/news start &
    fi
    ;;
  203.18.8.104)
# Скачать почту на мою домашнюю машину при установлении связи
# Маршрутизация не требуется, так как мой домашний Ethernet обрабатывается IP
```

```
# masquerade и proxyarp маршрутизацией.
    /usr/sbin/sendmail -q &
    ;;
*)
esac
exit 0
```

В результате поднятия ppp линка с нашим университетским городком Newman и выполнения этого сценария, мы получаем таблицы маршрутизации со следующими записями (эта машина является также нашим PPP сервером, и обрабатывает нашу связь с Интернет). Я разнообразил комментарии в выводе, чтобы помочь объяснить, чем является каждая запись):

```
[root@kepler /root]# route -n
Kernel routing table
Destination      Gateway          Genmask          Flags MSS      Window Use Iface
# HOST маршрут на наш удаленный шлюз в Интернет
139.130.177.2    *                255.255.255.255 UH    1500    0       134 ppp4
# HOST маршрут на наш Newman сервер студенческого городка
202.12.126.1     *                255.255.255.255 UH    1500    0       82  ppp5
# HOST маршрут на мой домашний ethernet
203.18.8.104     *                255.255.255.255 UH    1500    0       74  ppp3
# две наши основные коммутируемые PPP линии
203.18.8.64      *                255.255.255.255 UH    552     0       0   ppp2
203.18.8.62      *                255.255.255.255 UH    552     0       1   ppp1
# специальный сетевой маршрут в Newman LAN студенческого городка
202.12.126.0     202.12.126.1    255.255.255.0   UG    1500    0       0   ppp5
# маршрут в наш локальный Ethernet (суперсеть из двух сетей класса C)
203.18.8.0       *                255.255.254.0   U     1500    0       1683 eth0
# маршрут в устройство loop back
127.0.0.0        *                255.0.0.0       U     3584    0       483 lo
# маршрут по умолчанию в Internet
default          139.130.177.2   *                UG    1500    0       3633 ppp4
```

23.4 Обработка email

Предыдущий раздел показывает, как обработать исходящую почту - просто, очисткой (flushing) почтовой очереди, если связь установлена.

Если вы делаете WAN связь, то вы можете договориться с сетевым администратором удаленной LAN, чтобы сделать то же самое. Например, настройка нашей стороны WAN связи с Университетским городком Newman в файле /etc/ppp/ip-up выглядит так:

```
#!/bin/bash
#
# Скрипт, который занимается проблемами маршрутизации по мере
# необходимости для pppd
# Только связь с Hedland требует этой обработки.
#
# Когда связь ppp появляется, этот скрипт вызывается со следующими параметрами
# $1 имя интерфейса, используемого pppd (например ppp3)
```

```

#      $2      имя устройства tty
#      $3      скорость устройства tty
#      $4      локальный IP адрес для интерфейса
#      $5      удаленный IP адрес
#      $6      параметр, указанный опцией 'ipparam' в pppd
#
case "$5" in
    203.18.8.4)
        /usr/sbin/sendmail -q
        ;;
    *)
esac
exit 0

```

Если однако вы имеете с вашим ISP только динамический IP адрес по PPP связи, вы должны получать ваш email с аккаунта на машине вашего ISP. Это обычно делается с использованием POP (Протокол Почтового отделения). Этот процесс может быть выполнен, используя программу 'popclient', - а ip-up скрипт может автоматизировать этот процесс! Просто создайте сценарий /etc/ppp/ip-up, который содержит соответствующий вызов popclient. Для моей портативной ЭВМ, на которой работает Red Hat Linux (которую я беру в любые путешествия), это

```
popclient -3 -c -u hartr -p <password> kepler.hedland.edu.au |formail -s procmail
```

Вы могли бы использовать slurp или что-то еще, чтобы сделать это же самое для новостей и т.д. Помните, скрипт - это только стандартный скрипт bash и может использоваться так, чтобы автоматизировать ЛЮБУЮ функцию, которую нужно выполнять каждый раз при установлении соответствующей PPP связи.

24 Использование /etc/ppp/ip-down

Вы можете создать скрипт, который будет выполняться, как только связь будет завершена. Он называется /etc/ppp/ip-down. Он может использоваться, чтобы отменить что-нибудь, что вы сделали в соответствующем скрипте /etc/ppp/ip-up.

25 Маршрутизация в LAN

Если вы соединены с LAN, но также хотите использовать PPP на вашей персональной машине Linux, то вы должны адресовать одни пакеты в LAN (через ваш интерфейс ethernet), а другие на удаленный PPP сервер и дальше него (в Интернет).

Этот раздел не пытается обучить вас маршрутизации - он имеет дело только с простым, специальным случаем (статической) маршрутизации!

Я настойчиво прошу вас прочесть Руководство Сетевого Администратора Linux (NAG), если вы НЕ знакомы с маршрутизацией. Также книга O'Reilly "Сетевая Администрация TCP/IP" раскрывает эту тему в очень понятной форме.

Основное правило статической маршрутизации - ЗАДАННЫЙ ПО УМОЛЧАНИЮ маршрут должен быть тот, который указывает на НАИБОЛЕЕ число сетевых адресов. Для других сетей, введите специфические маршруты в таблицу маршрутизации.

Единственная ситуация, которую я собираюсь раскрыть здесь - это, когда ваш Linux PC находится в LAN, которая не соединена с Интернет - и вы хотите позвонить наружу в Интернет для персонального использования, сохраняя соединение с LAN.

Прежде всего удостоверитесь, что ваш маршрут Ethernet установлен так, чтобы специфические сетевые адреса были доступны в вашей LAN НЕ по маршруту по умолчанию!

Проверьте это, дав команду route. Вы должны увидеть что-то вроде следующего:

```
[root@hwin /root]# route -n
Kernel routing table
Destination      Gateway          Genmask         Flags MSS      Window Use Iface
loopback         *               255.255.255.0  U    1936    0        50 lo
10.0.0.0         *               255.255.255.0  U    1436    0       565 eth0
```

Если ваш интерфейс ethernet (eth0) указывает на маршрут по умолчанию, (первый столбец будет показывать "default" в строке eth0), то вы должны изменить ваши скрипты инициализации ethernet, чтобы заставить его указать на специфические сетевые адреса раньше, чем на заданный по умолчанию маршрут (проконсультируйтесь с Net2 HOWTO и NAG).

Это позволит rppd устанавливать ваш заданный по умолчанию маршрут как показано ниже:

```
[root@hwin /root]# route -n
Kernel routing table
Destination      Gateway          Genmask         Flags MSS      Window Use Iface
10.144.153.51   *               255.255.255.255 UH   488     0         0 ppp0
127.0.0.0       *               255.255.255.0  U    1936    0         50 lo
10.1.0.0        *               255.255.255.0  U    1436    0       569 eth0
default         10.144.153.51  *               UG   488     0         3 ppp0
```

Как вы видите, мы имеем главный маршрут к PPP серверу (10.144.153.51) через rpp0 и также заданный по умолчанию сетевой маршрут, который использует PPP сервер как шлюз.

Если ваши установки должны быть более сложными, чем эта - читайте документацию по маршрутизации, упомянутую выше и проконсультируйтесь с экспертом на вашем сайте!

Если в вашей LAN уже есть маршрутизатор, то вы уже будете иметь шлюз в глобальные сети, доступные вашему сайту. ТЕМ НЕ МЕНЕЕ вы должны указать ваш заданный по умолчанию маршрут на интерфейс PPP - и сделайте другие маршруты специфическими для сетей, которые они обслуживают.

25.1 Обратите внимание на защиту

Когда вы настраиваете компьютер с Linux в существующей LAN, чтобы подключиться к Интернет, то вы потенциально открываете вашу всю LAN Интернет - и взломщикам, которые постоянно находятся там. Прежде, чем вы сделаете это, я настойчиво прошу вас проконсультироваться с вашим сетевым администратором и стратегией защиты сайта. Если ваше PPP соединение с Интернетом будет использовано при успешной атаке на ваш сайт, то на вас по крайней мере обрушится гнев ваших пользователей, сетевых и системных администраторов. Также для вас могут возникнуть более серьезные проблемы!

Прежде, чем вы соедините LAN с Интернет, вы должны рассмотреть вопросы защиты даже для ДИНАМИЧЕСКОГО соединения - следовательно сначала обратитесь к O'Reilly "Построение файрволов Интернет"!

26 Установка PPP сервера

Как уже упомянуто, имеется много способов это сделать. Как я говорил, имеется способ, которым это делаю я (использование многопортовой платы Cyclades) и циклический пулл телефонных линий.

Если вам не нравится способ, который я представляю здесь, пожалуйста, не стесняйтесь идти собственным путем. Мне бы хотелось включить дополнительные методы в будущие версии HOW-TO. Так что, пожалуйста, вышлите мне ваши комментарии и способы!

Пожалуйста, обратите внимание, что этот раздел касается только установки Linux в качестве PPP сервера. Я (даже) не предполагаю включать информацию о установке специальных терминальных серверов итп.

Также я все-таки экспериментирую с теневыми паролями (но сделаю так когда-нибудь).

Информация в настоящее время не представлена, и следовательно никакие звоночки и свистки, которые требуются для shadow suite, не включаются.

26.1 Компиляция ядра

Примените все ранее высказанные комментарии о компиляции ядра и версиях ядра и соответствующих версий rppd. В этом разделе предполагается, что вы читали предыдущие разделы этого документа!

Для PPP сервера, Вы ДОЛЖНЫ включить IP forwarding в вашем ядре. Вы можете также захотеть включить другие возможности (типа IP fire wall, учета пользователей и т.д и т.д).

Если вы используете многопортовую последовательную плату, то вы очевидно также включили необходимые драйверы в ваше ядро!

26.2 Краткий обзор серверной системы

Мы предлагаем аккаунты коммутируемого PPP (и SLIP) и shell, использующие одну и ту же пару имени/пароля пользователя. Это имеет те преимущества (для нас), что для пользователя требуется только один аккаунт и можно использовать это для всех типов связей.

Поскольку мы - образовательная организация, мы не надзираем за доступом нашего персонала и студентов, и также не должны волноваться за учет пользователей и проблемах поддержки.

Мы эксплуатируем firewall между нашим сайтом и Интернетом, и это ограничивает некоторым пользователям доступ к телефонным линиям изнутри нашим (Интернет) firewall'ом (по довольно очевидным причинам, подробности наших других firewall'ов не рассматриваются и несоответствуют в любом случае).

Процесс, который пользователь проходит для установки PPP связи с нашим сайтом (конечно, если только ему разрешен доступ):

- Позвонить на наш циклический пулл (это единственный номер телефона, который соединяется с банком модемов - при звонке включается первый свободный модем).
- Вход в систему, если имя и пароль пользователя указаны правильно.
- В командной строке shell выдать команду rpp, чтобы запустить PPP на сервере.
- Запуск PPP на их PC (будь это Windows, DOS, Linux MAC OS или еще что) является проблемой пользователя).

Сервер использует индивидуальные файлы /etc/ppp/options.ttyXX для каждого модемного порта, который устанавливает динамический удаленный IP адрес. Сервер использует rppd маршрутизацию для удаленных клиентов (установлено соответствующей опцией rppd). Это устраняет потребность в routed или gated.

Когда пользователь "бросает трубку" на своем конце, rppd обнаруживает это и заставляет модем тоже "положить трубку", одновременно закрывая связь PPP.

26.3 Состав программного обеспечения

Вы нужно будет следующее программное обеспечение:

- Linux, правильно скомпилированный, чтобы были включены необходимые опции.

- соответствующая версия rpppd для вашего ядра.
- программа 'getty', которая интеллектуально обрабатывает модемную связь. Мы используем getty_ps2.0.7, но вроде mgetty лучше. Я понимаю, что mgetty может обнаружить звонок, который использует pap/chap (pap - стандарт для Windows95) и вызвать rpppd автоматически, но однако я должен исследовать это.
- Функционирующий сервер доменных имен (DNS), который доступен вашим модемным пользователям.

Вы действительно должны запустить собственный DNS, если это возможно ...

26.4 Установка стандартного (доступ через shell) коммутируемого доступа.

Прежде, чем вы сможете установить ваш PPP сервер, ваш Linux box должен быть способен обрабатывать стандартный доступ по коммутируемым линиям.

Этот howto не покрывает установку этой системы. Пожалуйста см. документацию по выбранному вами getty и Serial-HOWTO для информации об этом.

26.5 Установка файлов PPP опций

Вам нужно будет установить полный /etc/ppp/options с общими опциями для всех портов, принимающих внешние звонки. Опции, которые мы используем:

```
asynmap 0
netmask 255.255.254.0
proxyarp
lock
crtscts
modem
```

Примечание - мы НЕ используем (очевидную) маршрутизацию - и в частности, там нет опции defaultroute. Причина этого в том, что все, что вам (как PPP серверу) требуется сделать, это направить пакеты от внешнего ppp клиента через ваш LAN/INTERNET и маршрутизации пакетов клиента вашей LAN наружу.

Все, что для этого нужно - чтобы хост маршрутизировал клиентскую машину и использование опции rpppd 'proxyarp'.

Опция 'proxyarp' устанавливает (сюрприз) прокси ARP запись в ARP таблице PPP сервера, которая в основном говорит 'посылать все пакеты, предназначенные для PPP клиента ко мне'. Это самый простой способ установить маршрутизацию одному PPP клиенту, но вы не сможете использовать его, если вы маршрутизируете две LAN - вы должны добавить соответствующие сетевые маршруты, которые не могут использовать прокси ARP.

Вы наверняка пожелаете обеспечить динамическое распределение IP адресов для ваших модемных пользователей. Вы можете сделать это распределением IP адресов для каждого порта входящего модемного соединения. Теперь, создайте /etc/ppp/options.ttyXX для каждого порта входящего модемного соединения.

В них просто поместите локальный (серверный) IP адрес и IP адрес, который должен использоваться для этого порта. Например

```
kepler:slip01
```

В частности, обратите внимание, что вы можете использовать допустимые главные имена машин в этом файле (я нахожу, что напрмер я помню IP адреса только критических машин и устройств в моих сетях - имена легче запомнить)!

26.6 Настройка rppd, чтобы он был (успешно) доступен пользователям для запуска

Так как запуск rpp связи подразумевает конфигурирование устройства ядра (сетового интерфейса) и управление ядром, таблицы маршрутизации, требуются специальные привилегии - фактически полные привилегии root.

К счастью, rppd был разработан достаточно 'безопасным' для выполнения установки uid в root. Так что вам нужно будет сделать

```
chmod u+s /usr/sbin/pppd
```

когда вы выведете список файлов, то должно появиться

```
-rwsr-xr-x  1 root    root      74224 Apr 28 07:17 /usr/sbin/pppd
```

Если вы не сделаете это, то пользователи не смогут установить связь по rpp.

26.7 Установка глобального псевдонима для rppd

Для того, чтобы облегчить работу нашим модемным PPP пользователям, мы создадим глобальный псевдоним (в /etc/bashrc) так, чтобы одна простая команда запускала rpp на сервере, как только они регистрируются в системе.

Это выглядит примерно так

```
alias rpp="exec /usr/sbin/pppd -detach"
```

Что это значит

- `exec` : это означает заменить запущенную программу (в этом случае shell) программой, которая указана.
- `rppd -detach` : запустить rppd, и НЕ переводить в фоновый режим.

Это гарантирует, что когда rppd завершит работу, то не останется никаких других процессов. Когда пользователь регистрируется по этому способу, при выполнении команды 'w' он появляется как

```
 6:24pm up 3 days,  7:00,  4 users,  load average: 0.05, 0.03, 0.00
User      tty          login@  idle   JCPU   PCPU   what
hartr    ttyC0        3:05am  9:14          -
```

И это - все ... Я рассказал вам о простом, основном PPP сервере!

27 Использование PPP по нульмодемному (прямому последовательному) кабелю

Это очень просто - модема нет, так что дело значительно упрощается.

Прежде всего выберите одну из машин как 'сервер', установив getty на последовательный порт, так что вы можете проверить, что связь имеется, используя minicom, чтобы обратиться к последовательному порту на 'клиенте'.

Как только связь заработает, вы можете удалять getty если вы НЕ хотите удостовериться, что соединение работает, используя пару имя/пароль пользователя, как для модемного соединения. Поскольку вы имеете 'физическое управление' обеими машинами, я буду предполагать, что вы не хотите делать это.

Теперь, на сервере удалите getty, и удостоверьтесь, что вы имеете последовательные порты на обеих машинах, правильно сконфигурированные, используя 'setserial'.

Все, что вы должны сделать теперь, это запустить rppd на обеих системах. Я буду предполагать, что соединение использует /dev/ttyS34 на обеих машинах. Так, на обеих машинах выполните команду:

```
pppd -detach crtscts lock <local IP>:<remote IP> /dev/ttyS3 38400 &
```

Это установит связь - но пока еще у вас нет определенной маршрутизации. Вы можете проверить связь, пингуя обе машины. Если это работает, прекратите связь, уничтожив один из процессов rppd.

Какая вам нужна маршрутизация - это конечно зависит от того, что вы пытаетесь сделать. В общем случае, одна из машин будет соединена с ethernet (и внешним миром), и тогда требуется маршрутизация такая же, как для PPP сервера и клиента.

Так на машине оборудованной Ethernet, rppd команда будет

```
pppd -detach crtscts lock proxyarp <local IP>:<remote IP> /dev/ttyS3 38400 &
```

а на другой машине

```
pppd -detach crtscts lock defaultroute <local IP>:<remote IP> /dev/ttyS3 38400 &
```

Если вы связываете две сети (используя последовательную связь!) или реализуете более сложные требования маршрутизации, то вы можете использовать /etc/ppp/ip-up точно тем же образом, как упомянуто выше в этом документе.

Robert Hart Port Hedland, Western Australia Melbourne, Victoria, Australia August/October 1996 January/March 1997