

SMB HOWTO

David Wood, dwood@plugged.net.au, перевод Alex Ott ott@phtd.tpu.edu.ru v1.3, 20 апреля 2000

Это SMB HOWTO. Этот документ описывает использование в Linux, с помощью Samba, протокола Server Message Block (SMB), который иногда также называется протоколом Session Message Block (SMB), протоколом NetBIOS или протоколом LanManager.

Содержание

1	Лицензия	1
2	Введение	2
3	Дополнительная информация	2
4	Установка	3
5	Запуск демонов	4
6	Основная настройка (/etc/smb.conf)	5
7	Организация доступа к дискам Linux для машин под Windows	7
8	Доступ к SMB-ресурсам из Linux	9
9	Доступ к принтеру Linux для Windows-машин	11
10	Доступ к принтеру Windows с машин работающих под Linux	13
11	Резервирование данных с машин с Windows на машину с Linux	20
12	Использование Samba между маршрутизируемыми сетями	24
13	Авторские права	25
14	Благодарности Специальные благодарности Andrew Tridgell (tridge@linuxcare.com) за начало и направление проекта Samba и за сохранение этого документа в подлиннике.	25

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

1 Лицензия

Copyright (c) 2000 David Wood.

Разрешается копировать, распространять и/или изменять данный текст, под действием правил описанных в GNU Free Documentation License, Версии 1.1 или дальнейших, опубликованных Free Software Foundation; без постоянных разделов, без текстов обложек. Копия лицензии доступна по адресу <http://www.gnu.org/copyleft/fdl.html> <<http://www.gnu.org/copyleft/fdl.html>>.

2 Введение

Это SMB HOWTO. Этот документ описывает использование в Linux с помощью Samba протокола Server Message Block (SMB), который иногда также называется протоколом Session Message Block (SMB), протоколом NetBIOS или протоколом LanManager. Хотя этот документ и предназначен для Linux, но Samba может работать на большинстве Unix-подобных операционных систем.

Этот документ сопровождается David Wood (dwood@plugged.net.au). Добавления, изменения или исправления могут быть посланы ему для включения в следующий выпуск.

Большое количество документации по Samba доступно на WWW-сервере Samba, который находится по адресу <http://www.samba.org/> <<http://www.samba.org/>>. Там ужасающее количество информации; пожалуйста посмотрите ее до того, как будете просить помощи! Вы также можете использовать группу новостей comp.protocols.smb.

Протокол SMB используется Microsoft Windows 3.11, NT и 95/98 для организации доступа к дискам и принтерам. Используя набор утилит Samba от Andrew Tridgell (Andrew.Tridgell@anu.edu.au), UNIX-машины (включая Linux) могут организовывать доступ к дискам и принтерам для Windows-машин. Утилиты smbfs написанные Paal-Kr. Engstad (engstad@intermetrics.com) и Volker Lendecke (lendecke@namu01.gwdg.de) дают возможность машинам с Unix монтировать SMB-ресурсы доступные на машинах с Windows или Samba.

Четыре основные вещи, которые можно делать с помощью Samba:

1. давать доступ к дискам Linux Windows-машинам.
2. получать доступ к SMB-ресурсам с машин под Linux.
3. давать доступ к принтерам Linux для Windows-машин.
4. получать доступ к принтерам Windows из Linux-систем.

Обо всем этом рассказывается в этом документе, плюс также еще некоторые детали.

Отказ от ответственности: Процедуры и программы либо работают для автора, либо было сообщено, что они работают для людей, которые разработали их. Различные конфигурации могут не работать с информацией данной здесь. Если вы столкнулись с такой ситуацией, вы можете послать сообщение автору документа с пожеланиями для его улучшения.

Заметим, что для доступа к SMB-ресурсам с машин с Windows 3.x, на них должен быть установлен стек TCP/IP и библиотеки Win32s. Оба этих компонента доступны с сервера фирмы Microsoft (<http://www.microsoft.com> <<http://www.microsoft.com>>). Как нам сообщили во время написания данного документа, для того чтобы скачать стек TCP/IP-32 для Windows 3.x, Microsoft требует подписки на Microsoft Software Developers Network (MSDN). Поскольку это программное обеспечение используется свободно, то вы можете найти старые копии этого программного обеспечения у своих друзей.

3 Дополнительная информация

Этот документ делает попытку объяснить как настроить основные файловые и принтерные SMB-сервисы на Linux-машине. Samba является очень сложным и законченным пакетом. Незачем пытаться дублировать здесь всю документацию на Samba.

Для дополнительной информации смотрите следующие документы:

- Документация на Samba, доступная как часть дистрибутива Samba. Дистрибутив доступен по адресу: <ftp://ftp.samba.org/> <<ftp://ftp.samba.org/>>
- The Linux Printing HOWTO.
- The Print2Win Mini-HOWTO.

- Стандарт протокола для сервиса NetBIOS при использовании транспорта TCP/UDP.

RFC 1001

RFC 1001 - Концепции и методы.

RFC 1002

RFC 1002 - Подробные спецификации

4 Установка

Сначала, для того чтобы использовать Samba ваша машина должна находиться в одиночном ethernet-сегменте локальной сети, используя протокол TCP/IP. Samba не будет работать используя другие сетевые протоколы. Это в общем легко, поскольку Linux и Windows 95/98/NT поставляются с поддержкой TCP/IP. Однако если вы используете машины с Windows 3.X, то вам необходимо добавить поддержку TCP/IP. Один из самых распространенных вопросов, который я получаю— почему Samba "не работает", когда машины с Windows не используют TCP/IP.

Для того, чтобы установить TCP/IP на машине с Windows 95/98, выберите Control Panel | Network, затем добавьте и настройте Microsoft TCP/IP. В Windows NT, выберите Control Panel | Network | Protocols.

Для того, чтобы получить исходные тексты последней версии Samba, свяжите с указанным URL и выберите ближайший к вам сервер-зеркало: <ftp://ftp.samba.org/> <<ftp://ftp.samba.org/>>.

В большинстве случаев ваш дистрибутив Linux поставляется с пакетом, который содержит свежую версию Samba.

Следующие два демона необходимы для работы пакета Samba. Они обычно устанавливаются в `/usr/sbin` и запускаются либо при загрузке из системных скриптов, либо из `inetd`. Примеры скриптов даны в разделе **sec-daemons** (Запуск демонов).

<code>smbd</code>	(Демон SMB)
<code>nmbd</code>	(Обеспечивает поддержку сервера имен NetBIOS для клиентов)

Заметим, что сервис имен, предоставляемый демоном `nmbd` является отличным от сервиса имен, предоставляемого Domain Name Service (DNS). Сервис имен NetBIOS является сервисом имен в стиле Windows, используемым для SMB. Другими словами, имеющийся сервис имен DNS ничего не скажет о возможности Samba для нахождения имен машин.

Обычно следующие двоичные файлы Samba устанавливаются в директорию `/usr/bin`, хотя их расположение опционально.

<code>smbclient</code>	(Клиент SMB для UNIX-машин)
<code>smbprint</code>	(скрипт для печати на принтер на SMB-машине)
<code>smbprint.sysv</code>	(такое как выше, но для машин с SVR4 UNIX)
<code>smbstatus</code>	(Перечисляет текущие SMB-соединения для локальной машины)
<code>smbbrun</code>	('glue' скрипт для запуска приложений на SMB-машине)

Двоичные файлы относящиеся к поддержке файловой системы `smbfs` обсуждаются далее в этом документе.

Дополнительно с этим документом включен скрипт, названный 'print', который служит как полезная надстройка над скриптом `smbprint`.

Пакет Samba очень прост в установке. Просто получите исходные тексты по адресу, указанному выше, и прочитайте файл README в дистрибутиве. Также в дистрибутиве есть файл называемый `docs/INSTALL.txt`, который дает простой набор пошаговых инструкций.

Последующая установка, поместит демоны в `/usr/sbin` и двоичные файлы в `/usr/bin`. Установите справочные страницы в `/usr/local/man`.

При установке пакета Samba, вы должны указать в Makefile расположение файла конфигурации, `smb.conf`. Обычно это `/etc`, но вы можете поместить его куда угодно. Для этих указаний, мы будем предполагать, что вы указали расположение файла конфигурации как `/etc/smb.conf`, расположение файла протокола как `log file = /var/log/samba-log.%m` и блокировочной директории как `lock directory = /var/lock/samba`.

Установите файл конфигурации `smb.conf`. Перейдите в директорию, где была скомпилирована Samba. Посмотрите в поддиректории `examples/simple` и прочитайте файл `README`. Скопируйте файл `smb.conf` найденный в этой директории в директорию `/etc`. **БУДЬТЕ ОСТОРОЖНЫ!** Если у вас дистрибутив Linux с уже установленной Samba, у вас уже может быть файл конфигурации Samba в `/etc`. Вероятно вы можете начать работать с каким-то из них.

Если вы не хотите устанавливать файл конфигурации в директорию `/etc`, то поместите его куда вы хотите, а затем поместите символическую ссылку на него в директорию `/etc`:

```
ln -s /path/to/smb.conf /etc/smb.conf
```

5 Запуск демонов

Есть два демона SMB `/usr/sbin/smbd` и `/usr/sbin/nmbd`. В большинстве дистрибутивов Linux, они запускаются, останавливаются и перезапускаются через стартовый скрипт, расположенный в `/etc/rc.d/init.d/smb`, на который созданы символичные ссылки с соответствующих уровней запуска.

Если вы не будете использовать стандартные стартовые скрипты, то Вы можете запустить демоны Samba из `inetd` или как автономные процессы. Samba будет отвечать чуть быстрее когда она запущена как автономный процесс, чем в случае когда она запускается из `inetd`.

В некоторых случаях вы должны проверить наличие в файле `/etc/services` примерно таких строк:

<code>netbios-ns</code>	<code>137/tcp</code>	<code>nbns</code>
<code>netbios-ns</code>	<code>137/udp</code>	<code>nbns</code>
<code>netbios-dgm</code>	<code>138/tcp</code>	<code>nbdgm</code>
<code>netbios-dgm</code>	<code>138/udp</code>	<code>nbdgm</code>
<code>netbios-ssn</code>	<code>139/tcp</code>	<code>nbssn</code>

Убедитесь, что все они не закомментированы. В зависимости от вашего дистрибутива, вам может даже нужно будет добавить их в этот файл. Samba не сможет привязаться к соответствующим портам пока этих строк не будет в файле `/etc/services`.

Для запуска демонов из `inetd`, поместите следующие строки в конфигурационный файл `inetd`, `/etc/inetd.conf`:

```
# SAMBA NetBIOS services (for PC file and print sharing)
netbios-ssn stream tcp nowait root /usr/sbin/smbd smbd
netbios-ns dgram udp wait root /usr/sbin/nmbd nmbd
```

Затем перезапустите демон `inetd` выполнив команду:

```
kill -HUP 1 `cat /var/run/inetd.pid`
```

Для запуска демонов из системных стартовых скриптов, поместите следующий скрипт в файл `/etc/rc.d/init.d/smb` (для большинства дистрибутивов) и создайте на него символические ссылки с именами указанными в комментариях:

```
#!/bin/sh

#
# /etc/rc.d/init.d/smb - запускает и останавливает сервисы SMB
#
# Следующие файлы должны быть символическими ссылками на этот файл:
# symlinks:
#       /etc/rc.d/rc1.d/K35smb  (Убивает сервисы SMB при выключении
#       /etc/rc.d/rc3.d/S91smb  (Запускает сервисы SMB в многопользовательском режиме)
#       /etc/rc.d/rc6.d/K35smb  (Убивает сервисы SMB при перезагрузке)
#

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# See how we were called.
case "$1" in
  start)
    echo -n "Starting SMB services: "
    daemon smbd -D
    daemon nmbd -D
    echo
    touch /var/lock/subsys/smb
    ;;
  stop)
    echo -n "Shutting down SMB services: "
    killproc smbd
    killproc nmbd
    rm -f /var/lock/subsys/smb
    echo ""
    ;;
  *)
    echo "Usage: smb {start|stop}"
    exit 1
esac
```

Если при старте Samba вы получаете сообщение в котором говорится, что демон не может подключиться к порту 139, то вероятно у вас уже есть запущенные процессы Samba, которые не были завершены. Посмотрите список процессов (используя команду 'ps auxww | grep mbd') для того, чтобы определить есть ли еще запущенные сервисы Samba.

6 Основная настройка (/etc/smb.conf)

Настройка Samba в Linux (или других UNIX-машинах) контролируется единственным файлом, [/etc/smb.conf](#). Этот файл определяет к каким системным ресурсам вы хотите дать доступ для внешнего мира и какие ограничения вы хотите дать на использование этих ресурсов.

Так как следующие разделы посвящены организации доступа к дискам и принтерам Linux для Windows-машин, то файл [smb.conf](#), показанный в этом разделе является примером, исключительно в целях введения.

Не беспокойтесь сейчас о деталях. Следующие разделы введут основные концепции.

Каждый раздел файла начинается с заголовка раздела, такого как [global], [homes], [printers], и т.п.. Секция [global] определяет некоторые переменные, которые Samba будет использовать для определения доступа ко всем ресурсам.

Раздел [homes] позволяет удаленным пользователям иметь доступ к своим (и только своим) домашним директориям на локальной Linux-машине. Так что, если пользователи Windows попытаются подключиться к этому разделу со своих Windows-машин, то они будут подключены к своим персональным домашним директориям. Заметим, что для того чтобы сделать это, они должны быть зарегистрированы на Linux-машине.

Простой файл `smb.conf`, приведенный ниже, позволяет удаленным пользователям иметь доступ к их домашним директориям на локальной машине и писать во временную директорию. Для того, чтобы пользователи Windows могли увидеть эти ресурсы машина с Linux должна быть в локальной сети. Затем пользователи просто подключают сетевые диски с помощью Windows File Manager или Windows Explorer.

Заметим, что в следующих разделах будут даны дополнительные записи для этого файла, чтобы дать доступ к большему количеству ресурсов.

```

; /etc/smb.conf
;
; Убедитесь и перезапустите сервер после внесения изменений в этот
; файл, например:
; /etc/rc.d/init.d/smb stop
; /etc/rc.d/init.d/smb start

[global]
; Раскомментируйте эту строку, если вы хотите дать доступ пользователю "гость"
; guest account = nobody
log file = /var/log/samba-log.%m
lock directory = /var/lock/samba
share modes = yes

[homes]
comment = Home Directories
browseable = no
read only = no
create mode = 0750

[tmp]
comment = Temporary file space
path = /tmp
read only = no
public = yes

```

Написав новый файл `smb.conf`, полезно проверить его правильность. Вы можете проверить правильность написания файла `smb.conf`, используя утилиту 'testparm' (справочная страница: `testparm`); если `testparm` сообщает об отсутствии проблем, то `smbd` правильно загрузит файл настроек.

Полезный трюк: Если ваш сервер Samba имеет больше одного ethernet интерфейса, то `smbd` может подключиться к неправильному. Если это так, что вы можете явно заставить ее подключаться к нужному добавив строку в раздел [global] файла `/etc/smb.conf`:

```

interfaces = 192.168.1.1/24

```

здесь замените приведенный адрес на адрес вашего интерфейса ethernet. Значение "24" является правильным для сети класса C, но вам может понадобиться пересчитать это значение, если вы разделили ее на подсети. Это число относится к сетевой маске. Числа для других классов сетей приведены в IP-Masquerade mini-HOWTO.

Также существует GUI-утилита для настройки Samba: `GtkSamba`. Смотрите страницу по адресу <http://www.open-systems.com/gtksamba.html> <<http://www.open-systems.com/gtksamba.html>>.

7 Организация доступа к дискам Linux для машин под Windows

Как показано выше в файле smb.conf, организация доступа к дискам Linux для пользователей Windows является довольно простым делом. Однако, с помощью Samba вы можете контролировать это в большой степени. Здесь мы приведем несколько примеров:

Чтобы организовать доступ для всех, создайте копию раздела [tmp], приведенного выше, добавив что-то подобное в smb.conf:

```
[public]
  comment = Public Stuff
  path = /home/public
  public = yes
  writable = yes
  printable = no
```

Вы сейчас будете иметь возможность запустить Samba и просмотреть доступные ресурсы с Windows PC. Однако, недавно Microsoft сделала жизнь немного более трудной для тех, кто использует Samba. Windows 98, Windows NT (service pack 3 или выше) и поздние версии Windows 95 теперь по умолчанию используют шифрованные пароли. А Samba по умолчанию использует нешифрованные пароли. Вы не сможете просматривать содержимое серверов когда либо клиент либо сервер используют шифрованные пароли, потому что соединение не может быть сделано без регистрации.

Если у вас есть несоответствие типов паролей между клиентом и сервером и когда вы пытаетесь подключиться к ресурсу, то вы можете увидеть окно диалога в котором будет сказано примерно следующее: "You are not authorized to access that account from this machine".

Вы можете либо настроить Samba-сервер для использования шифрованных паролей, либо настроить машины с Windows, чтобы они использовали нешифрованные пароли.

Для того, чтобы заставить Windows работать с нешифрованными паролями SMB:

```
Windows 95/98
=====
```

```
Используя редактор registry (regedit), создайте ключ registry
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP
Добавьте значение типа DWORD:
Имя значения: EnablePlainTextPassword
Данные:          0x01.
```

```
Windows NT
=====
```

```
Используя редактор registry (regedit), создайте ключ registry
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Rdr\Parameters
Добавьте значение типа DWORD:
Имя значения: EnablePlainTextPassword
Значение:          0x01
```

```
Windows 2000
=====
```

```
Using the registry editor (regedit), create the registry setting
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkStation\Parameters
Add a new DWORD value:
Value Name: EnablePlainTextPassword
Data: 0x01
```

После того, как вы сделали эти изменения, перезагрузите машину с Windows и попытайтесь подключить сетевой диск к серверу Samba. Это должно работать, поскольку сервер Samba использует пароли в нешифрованном виде.

Для того, чтобы настроить Samba для использования зашифрованных паролей:
В раздел [global] файла /etc/smb.conf, добавьте следующие строки:

```
encrypt passwords = yes
smb passwd file = /etc/smbpasswd
```

Мы вам настоятельно рекомендуем прочитать файлы ENCRYPTION.txt, Win95.txt и WinNT.txt в документации по Samba до того как вы будете делать это!

Если ваши клиенты и сервер используют зашифрованные пароли, то вы не сможете просмотреть списки доступных ресурсов до тех пор пока начальное соединение не будет сделано с правильными параметрами входа. Для того чтобы сделать первоначальное соединение, введите вручную имя ресурса в диалоговом окне Windows File Manager или Explorer, имя должно быть в форме '\\<hostname>\<sharename>'. Подключитесь к серверу и правильными для данного сервера именем и паролем.

Если вы подозреваете, что ваш сервис имен NetBIOS неправильно настроен (возможно потому, что вы получаете сообщение об ошибке 'host not found (машина не найдена)' при попытке подключения), то попытайтесь использовать IP-адрес сервера: '\\<host ip address>\<sharename>'.
Для того, чтобы имена отображались правильно, вы можете понадобиться установить некоторые настройки в соответствующем разделе ресурса. Это работает с клиентами под управлением 95/98/NT, но может быть необходимо будет изменить эти настройки, если вы используете клиентов с Windows 3.X:

```
; Mangle case = yes seems to give the correct filenames
; for Win95/98/NT.
mangle case = yes

; If samba is case sensitive when looking for files
case sensitive = no

; Default case of files that are created
default case = lower

; Preserve case for all filenames
preserve case = yes

; Preserve case for dos (8.3) filenames
short preserve case = no
```

Замечание: Значение после знака / это ссылка на сетевую маску. Для сети класса C этим значением будет "24". Для дополнительной информации о том как выполнять вычисления для подсетей, вы можете посмотреть информацию по адресу <http://www.ralphb.net/IPSubnet/> <<http://www.ralphb.net/IPSubnet/>>.

Конечно настройка Samba более полна, чем мы привели здесь, но это даст вам возможность начать работать. Если вы хотите сделать что-то более сложное, то я рекомендую вам сходить на сервер Samba, упомянутый выше.

Чтобы сделать эту директорию доступной для чтения всеми, но разрешить писать в нее только людям из группы staff, измените запись как показано в следующем примере:

```
[public]
comment = Public Stuff
path = /home/public
public = yes
writable = yes
printable = no
write list = @staff
```

Для описания других приемов для изменения доступа к дискам, смотрите документацию на Samba и справочные страницы.

8 Доступ к SMB-ресурсам из Linux

Машины с Linux (UNIX) могут также просматривать и монтировать SMB-ресурсы. Заметьте, что это может быть сделано когда сервером является либо машина с Windows, либо с сервером Samba! Программа клиента SMB для UNIX-машин включена в дистрибутив Samba. Она обеспечивает ftp-подобный интерфейс командной строки. Вы можете использовать эту утилиту для переноса файлов между 'сервером' под управлением Windows и клиентом под Linux.

Большинство дистрибутивов Linux также включают полезный пакет smbfs, который позволяет монтировать и размонтировать SMB-ресурсы. Больше информации о smbfs можно получить дальше.

Для того, чтобы увидеть какие ресурсы доступны на данной машине, выполните команду:

```
/usr/bin/smbclient -L host
```

где 'host' это имя машины, которую вы хотите видеть. Эта команда вернет список имен 'сервисов' - т.е. имен дисков или принтеров к которым может быть получен доступ. До тех пор, пока SMB сервер не будет настроен без управления доступом, он будет запрашивать у вас пароль. Введите в ответ на запрос пароль для пользователя 'гость (guest)' или ваш персональный пароль на этой машине.

Например:

```
smbclient -L zimmerman
```

Вывод этой команды должен выглядеть примерно так:

```
Server time is Sat Aug 10 15:58:27 1996
Timezone is UTC+10.0
Password:
Domain=[WORKGROUP] OS=[Windows NT 3.51] Server=[NT LAN Manager 3.51]

Server=[ZIMMERMAN] User=[] Workgroup=[WORKGROUP] Domain=[]

      Sharename      Type      Comment
      -----      -
ADMIN$             Disk      Remote Admin
public             Disk      Public
C$                 Disk      Default share
IPC$               IPC       Remote IPC
OReilly            Printer   OReilly
print$             Disk      Printer Drivers
```

This machine has a browse list:

Server	Comment
-----	-----
HOPPER	Samba 1.9.15p8
KERNIGAN	Samba 1.9.15p8
LOVELACE	Samba 1.9.15p8
RITCHIE	Samba 1.9.15p8
ZIMMERMAN	

Browse list показывает другие SMB сервера в сети с доступными ресурсами.

Для использования клиента, выполните следующую команду:

```
/usr/bin/smbclient service <password>
```

где 'service' — имя машины и сервиса. Например, если вы пытаетесь обратиться к директории, которая доступна под именем 'public' на машине названной 'zimmerman', то имя сервиса должно звучать как `\\zimmerman\public`. Однако в следствии ограничений оболочки, вам необходимо спрятать обратный слэш, так что в итоге это командная строка выглядеть следующим образом:

```
/usr/bin/smbclient \\\\zimmerman\public mypasswd
```

где 'mypasswd' — символьная строка вашего пароля.

Вы получите приглашение smbclient:

```
Server time is Sat Aug 10 15:58:44 1996
Timezone is UTC+10.0
Domain=[WORKGROUP] OS=[Windows NT 3.51] Server=[NT LAN Manager 3.51]
smb: \>
```

Напечатайте 'h' чтобы получить помощь об использовании smbclient:

```
smb: \> h
ls          dir          lcd          cd           pwd
get         mget        put          mput        rename
more        mask        del          rm           mkdir
md          rmdir      rd           prompt      recurse
translate  lowercase  print       printmode   queue
cancel     stat        quit        q           exit
newer      archive    tar         blocksize   tarmode
setmode    help       ?           !
smb: \>
```

Если вы умеете использовать ftp, то вам не будут нужны справочные страницы о smbclient.

Хотя вы можете использовать smbclient для тестирования, вы скоро устанете использовать его в работе. Для работы вам скорее всего понадобится пакет smbfs. Smbfs поставляется с двумя простыми утилитами, smbmount и smbmount. Они работают подобно mount и umount для SMB-ресурсов.

Одно важное замечание: Чтобы использовать эти утилиты, у вас должна быть поддержка smbfs вкомпилированная в ядро!

Следующий пример показывает типичное использование smbmount для монтирования SMB-ресурса, названного "customers" с машины с именем "samba1":

```
[root@postel]# smbmount "\\\\samba1\customers" -U rtg2t -c 'mount /customers -u 500 -g 100'
Added interface ip=192.168.35.84 bcast=192.168.255.255 nmask=255.255.0.0
Got a positive name query response from 192.168.168.158 ( 192.168.168.158 )
Server time is Tue Oct 5 10:27:36 1999
Timezone is UTC-4.0
Password:
Domain=[IPM] OS=[Unix] Server=[Samba 2.0.3]
security=user
```

Запуск команды mount покажет вам, что ресурс смонтирован, точно также как и через экспорт NFS:

```
[root@postel]# mount
/dev/hda2 on / type ext2 (rw)
none on /proc type proc (rw)
none on /dev/pts type devpts (rw,mode=622)
//SMBAL/CUSTOMERS on /customers type smbfs (0)
```

Пожалуйста прочитайте справочные страницы smbmount и smbmount для дополнительной информации о описанных выше операциях.

9 Доступ к принтеру Linux для Windows-машин

Для организации доступа к Linux-принтеру с Windows машин, вам необходимо убедиться, что принтер работает под Linux. Если вы можете печатать под Linux, то организация доступа к принтеру будет очень проста.

Заметьте, что для того, чтобы печатать пользователи Windows должны быть зарегистрированы на сервере Linux/Samba. Windows 95/98 будет пытаться аутентифицироваться на сервер печати с тем именем и паролем, который использовался для входа на машину с Windows. Это значит, что если вы нажали на 'Cancel (Отмена)' при входе в Windows, то вы не сможете печатать, или подключаться к SMB сервисам! Windows NT позволяет явно задавать имя и пароль при подключении к принтеру. Смотрите Printing HOWTO для дополнительной информации о настройке печати.

Добавьте настройку принтера в ваш файл smb.conf:

```
[global]
printing = bsd
printcap name = /etc/printcap
load printers = yes
log file = /var/log/samba-log.%m
lock directory = /var/lock/samba

[printers]
comment = All Printers
security = server
path = /var/spool/lpd/lp
browseable = no
printable = yes
public = yes
writable = no
create mode = 0700

[ljet]
security = server
path = /var/spool/lpd/lp
printer name = lp
writable = yes
public = yes
printable = yes
print command = lpr -r -h -P %p %s
```

Убедитесь, что путь к принтеру (в этом случае для [ljet]) соответствует буферной директории, указанной в файле /etc/printcap!

Строки:

```
printcap name = /etc/printcap
load printers = yes
```

контролируют должны ли все принтера, перечисленные в /etc/printcap по умолчанию быть загружены. Если вы сделаете это, то нет причины для настройки каждого принтера в отдельности. Раздел [printers] указывает настройки для принтеров, которые вы хотите определить явно. Если используемая вами подсистема печати не работает этим образом (BSD), то вам необходимо настроить фальшивый файл printcap (или используйте 'print command', смотрите ниже). Для дополнительной информации о системе printcap смотрите Printing-HOWTO.

Полезным приемом для проверки сетевых соединений является изменение команды печати на:

```
print command = cp %S /tmp/print.%.%S
```

Итоговый файл может быть проанализирован.

ЗАМЕЧАНИЕ: Существуют некоторые проблемы с доступом к принтерам на UNIX-машинах для машин с Windows NT используя Samba. Одна из проблем в том, что NT неправильно видит сетевой принтер. Для решения этой проблемы, смотрите замечание в файле [docs/WinNT.txt](#) дистрибутива Samba. Другая проблема связана с проблемой пароля. Смотрите комментарий в том же самом файле.

Oleg L. Machulskiy (machulsk@shade.msu.ru) предложил более удобную для использования команду печати, чем в вышеприведенном примере:

```
print command = smb2ps %s | lpr -r -h -P %p
```

где 'smb2ps' является скриптом, который преобразует буферный файл, переданный из под Windows в обычно используемый файл Postscript. Он должен удалять первые три и последние две строки, потому что эти строки содержат некоторые коды PJL или PCL.

Такой подход нужен только если ваша машина с Windows печатает PCL, а не настоящий Postscript. Я обнаружил, что Windows 95/98/NT не имеют общего драйвера Postscript, но драйвер принтера "Digital turbo Printserver 20" может работать как хороший драйвер Postscript для большинства случаев. Я также слышал, что для этих же целей может работать драйвер "Apple LaserWriter II NTX".

Если вы создали буферный каталог для печати, вместо использования созданных при установке дистрибутива Linux, то будьте внимательны с правами доступа! Neil Fraser (neilf@necan.co.za) предложил права доступа к буферному каталогу (в нашем случае это, /var/spool/lpd/lpr) сделать равными 4755 (заметьте, что установлен suid-бит). Это работает в его конфигурации, если владельцем каталога является пользователь 'root', а группа равна 'lp'.

Jeff Stern (jstern@eclectic.ss.uci.edu) написал, что он установил права доступа к каталогу равными 777, для того, чтобы обычные пользователи могли печатать, хотя он заметил, что он также добавил пользователей в группу 'lp'. Это решение пригодно для локальных системных администраторов; если безопасность печати является предметом споров, то не используйте данное решение. В домашних условиях, вы вероятно захотите, чтобы все могли печатать.

Dr. Michael Langner (langner@fiz-chemie.de) указал, что проблема прав доступа на запись на дерево директорий /var/spool/lpd/ может быть избегнута, использованием чего-то подобного "path=/tmp" и "print command = lpr -r -P%p %s".

Иногда возникает ошибка разбора Postscript при печати Postscript с машины под Windows, что вызывает печать лишних страниц в конце каждого задания. Последняя страница всегда будет содержать сообщение "%[Lastpage]%" в заголовке. Это кажется случается только на машинах с Windows 95 и 98 и это происходит из-за того, что Postscript является плохо-сформированным.

Одним из способов обработки этой ошибки — использовать скрипт для удаления куска плохого Postscript из задания печати. Другим способом является нахождение лучшего драйвера Postscript для Windows. Вероятно лучший способ для нас это использовать LPRng вместо Postscript для печати на сервер Samba.

Erik Ratcliffe (erik@caldera.com), Caldera, сообщил, что использование LPRng означает, что любой драйвер принтера может быть использован с машин под Windows. На сервере Samba, они используют запись в /etc/printcap, которая выглядит примерно так:

```
raw:\
    :rw:sh:
    :lp=/dev/lp1
    :sd=/var/spool/lpd/raw
    :fx=flp
```

LPRng не требует :\ в конце каждой строки. Запись о принтере все равно необходимо сделать в файле /etc/smb.conf для физического принтера. Строка команды печати нужна для использования записи "raw" из файла /etc/printcap и данные должны посылаться в двоичной форме. Попробуйте вот такую командную строку:

```
print command = lpr -b -Prw %s
```

Вам также может понадобиться установить буферизацию печати на Windows95 для печати прямо на принтер, вместо использования буферизации на сервере.

Если вы при печати постоянно получаете лишние страницы в конце задания с клиентов с Windows, то попробуйте добавить директиву "sf" в файл /etc/printcap. Это запретит прогон страниц, разделяющих задание, но не будет влиять на прогон страниц внутри документов.

10 Доступ к принтеру Windows с машин работающих под Linux

Для доступа к принтеру на Windows машине, вы должны сделать следующее:

1. ВБ должны иметь правильные записи в файле `/etc/printcap` и они должны соответствовать локальной структуре директорий (для буферной директории, и т.п.)
2. У вас должен быть скрипт `/usr/bin/smbprint`. Он поставляется вместе с исходными текстами Samba, но не со всеми двоичными дистрибутивами Samba. Его немного модифицированная копия обсуждается ниже.
3. Если вы хотите преобразовывать ASCII файлы в Postscript, вы должны иметь программу `nenscript`, или ее эквивалент. `nenscript` — это конвертер Postscript, он обычно устанавливается в директорию `/usr/bin`.
4. Вы можете захотеть сделать печать через Samba более легкой, используя программы-надстройки. Простой скрипт на perl, который обрабатывает ASCII, Postscript или преобразованный Postscript приведен ниже.
5. Вы также можете использовать MagicFilter для того, чтобы выполнить, описанное выше. Подробности о настройке MagicFilter приводятся ниже. MagicFilter имеет преимущества, потому что он знает как автоматически преобразовывать достаточно большое количество форматов файлов.

Запись в файле `/etc/printcap`, приведенном ниже, сделана для принтера HP 5MP на сервере Windows NT. Используются следующие поля файла `/etc/printcap`:

```

cm - комментарий
lp - имя устройства, открываемого для вывода
sd - директория спула принтера (на локальной машине)
af - файл учета пользования принтером
mx - максимальный размер файла (ноль -- без ограничений)
if - имя входного фильтра (скрипта)

```

Для более детальной информации о печати смотрите *Printing HOWTO* или справочные страницы по `printcap`.

```

# /etc/printcap
#
# //zimmerman/oreilly via smbprint
#
lp:\
    :cm=HP 5MP Postscript OReilly on zimmerman:\
    :lp=/dev/lp1:\
    :sd=/var/spool/lpd/lp:\
    :af=/var/spool/lpd/lp/acct:\
    :mx#0:\
    :if=/usr/bin/smbprint:

```

Убедитесь, что буферные директории и директория, используемая для учета пользования существуют и имеют право на запись. Убедитесь, что строка 'if' содержит правильный путь к скрипту smbprint (дан ниже) и убедитесь, что записи указывают на правильное устройство вывода (специальный файл /dev).

Далее идет сам скрипт smbprint. Он обычно находится в директории /usr/bin и написан Andrew Tridgell, человеком, который пакет создал Samba, насколько я знаю. Этот скрипт поставляется вместе с дистрибутивом исходного кода Samba, но отсутствует в некоторых бинарных дистрибутивах, так что я воссоздал его здесь.

Вы можете захотеть взглянуть на него более внимательно. Есть некоторые мелкие изменения, которые показали себя полезными.

```
#!/bin/sh -x

# Этот скрипт является входным фильтром для основанной на printcap
# печати на unix-машинах. Он использует программу smbclient для
# печати файла на указанный smb-сервер и сервис.
# Например вы можете иметь запись в printcap подобную этой
#
# smb:lp=/dev/null:sd=/usr/spool/smb:sh:if=/usr/local/samba/smbprint
#
# которая создает unix-принтер названный "smb", который будет
# печатать с помощью этого скрипта. Вам необходимо создать директорию
# спула /usr/spool/smb с соответствующими правами и владельцем

# Установите здесь сервер и сервис на который вы хотите печатать. В
# этом примере я имею PC с WfWg PC, названную "lapland", которая
# имеет экспортируемый принтер, называемый "printer" без пароля

#
# Далее скрипт был изменен hamilton@ecnz.co.nz (Michael Hamilton)
# так что сервер, сервис и пароль могут быть считаны из файла
# /usr/var/spool/lpd/PRINTNAME/.config
#
# Для того чтобы это работало запись в /etc/printcap должна
# включать файл учета использования (af=...):
#
#   cdcolour:\
#       :cm=CD IBM Colorjet on 6th:\
#       :sd=/var/spool/lpd/cdcolour:\
#       :af=/var/spool/lpd/cdcolour/acct:\
#       :if=/usr/local/etc/smbprint:\
#       :mx=0:\
#       :lp=/dev/null:
#
# Файл /usr/var/spool/lpd/PRINTNAME/.config должен содержать
#   server=PC_SERVER
#   service=PR_SHARENAME
#   password="password"
#
# Например,
#   server=PAULS_PC
#   service=CJET_371
#   password=""

#
# Debugging log file, change to /dev/null if you like.
#
```

```

logfile=/tmp/smb-print.log
# logfile=/dev/null

#
# The last parameter to the filter is the accounting file name.
#
spool_dir=/var/spool/lpd/lp
config_file=$spool_dir/.config

# Should read the following variables set in the config file:
#   server
#   service
#   password
#   user
eval `cat $config_file`

#
# Some debugging help, change the >> to > if you want to same space.
#
echo "server $server, service $service" >> $logfile

(
# NOTE You may wish to add the line `echo translate' if you want automatic
# CR/LF translation when printing.
    echo translate
    echo "print -"
    cat
) | /usr/bin/smbclient "\\\\$server\\$service" $password -U $user -N -P >> $logfile

```

Большинство дистрибутивов Linux поставляется с программой `nenscript` для преобразования ASCII документов в Postscript. Следующий скрипт на perl делает жизнь пользователя легче, обеспечивая простой интерфейс для печати используя `smbprint`.

```

Использование: print [-a|c|p] <filename>
    -a печатает <filename> как ASCII
    -c печатает <filename> отформатированный как исходный код
    -p печатает <filename> как Postscript
    Если опции не заданы, программа попытается определить
    тип файла и печатать соответственно

```

Используя `smbprint` для печати ASCII файлов, скрипт следит за длинными строками. Если возможно, этот скрипт разрывает длинную строку на пробеле (вместо разрыва в середине слова).

Форматирование исходного кода выполняется с помощью программы `nenscript`. Она берет ASCII-файл и форматирует его в 2 колонки с заголовком (дата, имя файла и т.п.). Эта программа также нумерует строки. Используя этот скрипт как пример, могут быть добавлены другие типы форматирования.

Postscript-документы являются уже отформатированы, так что они печатаются сразу.

```

#!/usr/bin/perl

# Скрипт:   print
# Авторы:  Brad Marshall, David Wood
#          Plugged In Communications
# Дата:    960808
#

```

```
# Используется для печати на сервис oreilly, который расположен на
# сервере zimmerman
# Назначение: Берет файлы разных типов как аргумент и обрабатывает
# их соответственно для передачи на скрипт печать Samba.
#
# В настоящее время поддерживаются типы файлов:
#
# ASCII      - Если длина строки длиннее чем $line_length символов, то
#              переносит строку на пробеле
# Postscript - Берет без обработки
# Code       - Форматирует в Postscript (используя nenscript), чтобы
#              отображать правильно (альбомный формат, шрифт и т.п.)
#
# Установить максимальную длину строки ASCII текста
$line_length = 76;

# Установить путь к скрипту печати Samba
$print_prog = "/usr/bin/smbprint";

# Установить путь и имя nenscript (конвертера ASCII-->Postscript)
$nenscript = "/usr/bin/nenscript";

unless ( -f $print_prog ) {
    die "Can't find $print_prog!";
}
unless ( -f $nenscript ) {
    die "Can't find $nenscript!";
}

&ParseCmdLine(@ARGV);

# DBG
print "filetype is $filetype\n";

if ($filetype eq "ASCII") {
    &wrap($line_length);
} elsif ($filetype eq "code") {
    &codeformat;
} elsif ($filetype eq "ps") {
    &createarray;
} else {
    print "Sorry..no known file type.\n";
    exit 0;
}
# Pipe the array to smbprint
open(PRINTER, "|$print_prog") || die "Can't open $print_prog: $!\n";
foreach $line (@newlines) {
    print PRINTER $line;
}
# Send an extra linefeed in case a file has an incomplete last line.
print PRINTER "\n";
close(PRINTER);
print "Completed\n";
exit 0;

# ----- #
#           Everything below here is a subroutine           #
```



```
# ----- #

sub ParseCmdLine {
    # Parses the command line, finding out what file type the file is

    # Gets $arg and $file to be the arguments (if the exists)
    # and the filename
    if ($#_ < 0) {
        &usage;
    }
    # DBG
    # foreach $element (@_) {
    #     print "$element" \n";
    # }

    $arg = shift(@_);
    if ($arg =~ /\-./) {
        $cmd = $arg;
    # DBG
    # print "\$cmd found.\n";

        $file = shift(@_);
    } else {
        $file = $arg;
    }

    # Defining the file type
    unless ($cmd) {
        # We have no arguments

        if ($file =~ /\.ps$/) {
            $filetype = "ps";
        } elsif ($file =~ /\.java$|\.c$|\.h$|\.pl$|\.sh$|\.csh$|\.m4$|\.inc$|\.html$|\.h
            $filetype = "code";
        } else {
            $filetype = "ASCII";
        }

        # Process $file for what type is it and return $filetype
    } else {
        # We have what type it is in $arg
        if ($cmd =~ /^-p$/) {
            $filetype = "ps";
        } elsif ($cmd =~ /^-c$/) {
            $filetype = "code";
        } elsif ($cmd =~ /^-a$/) {
            $filetype = "ASCII"
        }
    }
}

sub usage {
    print "
Использование: print [-a|c|p] <filename>
-a печатает <filename> как ASCII
-c печатает <filename> отформатированный как исходный код
-r печатает <filename> как Postscript

```

```
Если опции не заданы, программа попыбует определить
тип файла и печатать соответственно\n
";
exit(0);
}

sub wrap {
# Create an array of file lines, where each line is < the
# number of characters specified, and wrapped only on whitespace

# Get the number of characters to limit the line to.
$limit = pop(@_);

# DBG
#print "Entering subroutine wrap\n";
#print "The line length limit is $limit\n";

# Read in the file, parse and put into an array.
open(FILE, "<$file") || die "Can't open $file: $!\n";
while(<FILE>) {
    $line = $_;

    # DBG
    #print "The line is:\n$line\n";

    # Wrap the line if it is over the limit.
    while ( length($line) > $limit ) {

        # DBG
        #print "Wrapping...";

        # Get the first $limit +1 characters.
        $part = substr($line,0,$limit +1);

        # DBG
        #print "The partial line is:\n$part\n";

        # Check to see if the last character is a space.
        $last_char = substr($part,-1, 1);
        if ( " " eq $last_char ) {
            # If it is, print the rest.

            # DBG
            #print "The last character was a space\n";

            substr($line,0,$limit + 1) = "";
            substr($part,-1,1) = "";
            push(@newlines,"$part\n");
        } else {
            # If it is not, find the last space in the
            # sub-line and print up to there.

            # DBG
            #print "The last character was not a space\n";

            # Remove the character past $limit
            substr($part,-1,1) = "";
            # Reverse the line to make it easy to find
```

```

        # the last space.
        $revpart = reverse($part);
        $index = index($revpart, " ");
        if ( $index > 0 ) {
            substr($line,0,$limit-$index) = "";
            push(@newlines,substr($part,0,$limit-$index)
                . "\n");
        } else {
            # There was no space in the line, so
            # print it up to $limit.
            substr($line,0,$limit) = "";
            push(@newlines,substr($part,0,$limit)
                . "\n");
        }
    }
    push(@newlines,$line);
}
close(FILE);
}

sub codeformat {
    # Call subroutine wrap then filter through nenscript
    &wrap($line_length);

    # Pipe the results through nenscript to create a Postscript
    # file that adheres to some decent format for printing
    # source code (landscape, Courier font, line numbers).
    # Print this to a temporary file first.
    $tmpfile = "/tmp/nenscript$$";
    open(FILE, "|$nenscript -2G -i$file -N -p$tmpfile -r") ||
        die "Can't open nenscript: $!\n";
    foreach $line (@newlines) {
        print FILE $line;
    }
    close(FILE);

    # Read the temporary file back into an array so it can be
    # passed to the Samba print script.
    @newlines = ("");
    open(FILE, "<$tmpfile") || die "Can't open $file: $!\n";
    while(<FILE>) {
        push(@newlines,$_);
    }
    close(FILE);
    system("rm $tmpfile");
}

sub createarray {
    # Create the array for postscript
    open(FILE, "<$file") || die "Can't open $file: $!\n";
    while(<FILE>) {
        push(@newlines,$_);
    }
    close(FILE);
}
}

```

Теперь о применении MagicFilter. Спасибо Alberto Menegazzi (flash.egon@iol.it) за его информацию.

Alberto сообщил: ————— 1) Установите MagicFilter в /usr/bin/local с фильтрами для необходимых принтеров, но НЕ заполняйте записи в /etc/printcap, как предполагается в документации на MagicFilter.

2) Запишите в /etc/printcap примерно вот такую запись (Это сделано для моего принтера LaserJet 4L):

```
lp|ljet4l:\ :cm=HP LaserJet 4L:\ :lp=/dev/null:\ # or /dev/lp1 :sd=/var/spool/lpd/ljet4l:\ :af=/var/spool/lpd/ljet4l/acct:\ :sh:mx#0:\ :if=/usr/local/bin/main-filter:
```

Вы должны, объяснить, что устройство lp=/dev/... открывается для блокирования, так что для каждого удаленного принтера используется одно "виртуальное устройство".

Пример создания : touch /dev/ljet4l

3) Напишите фильтр /usr/local/bin/main-filter, с таким же образом предполагая использование ljet4l-filter вместо cat.

Вот так для меня.

```
#!/bin/sh logfile=/var/log/smb-print.log spool_dir=/var/spool/lpd/ljet4l ( echo "print - /usr/local/bin/ljet4l-filter ) | /usr/bin/smbclient "\\\SHIR\\HPLJ4N -P >> $logfile
```

P.S. : Это цитата из Print2Win mini-Howto о блокировании, а также о том, зачем создавать виртуальные принтера

—Начало здесь ——— Совет от Rick Bressler :

Хороший совет. Я использую нечто подобное. Вот один полезный совет, хотя он не является хорошей идеей:

```
:lp=/dev/null:\
```

lpг делает 'монопольное (exclusive)' открытие файла, который вы укажете в поле lp=. Он делает это для предотвращения попыток множества процессов печатать одновременно на одном и том же принтере.

Побочным эффектом этого, в вашем случае является то, что eng и colour не могут печатать одновременно, (обычно более или менее прозрачно, поскольку они вероятно печатают быстро и поскольку вы не замечаете, что они ставят задания в очередь), но любые другие процессы, которые пытаются записать в /dev/null не будут работать!

На однопользовательской системе, это вероятно не является большой проблемой. У меня имеется система с 50 принтерами. В этом случае это может быть проблемой.

Решение этой проблемы заключалось в создании устройства для каждого их принтеров. Например: touch /dev/eng.

Я модифицировал записи lp в файле printcap, приведенном выше, приняв во внимание пожелания Rick. Я сделал следующее:

```
# touch /dev/eng # touch /dev/colour
```

—Конец — —

11 Резервирование данных с машин с Windows на машину с Linux

Adam Neat (adamneat@ipax.com.au) предоставил следующий скрипт, который выполняет резервирование данных с машин с Windows machines на машину с Linux, используя утилиту smbclient. Adam сказал, что он использует этот скрипт для резервирования данных с машин с Windows 3.x и NT на диск DAT SCSI под Linux.

Adam не гордится стилем программирования, используемым в этом примере, но это работает. Как я хочу сказать: "Если это работает и это сделано тупо, то это не тупо".

Другой скрипт для сохранения данных с Windows, предоставлен Dan Tager (dtager@marsala.com) и приводится ниже. Скрипт присланный Dan также может сохранять данные с машин с Unix через rsh, хотя он может легко быть изменен для использования ssh.

В этом скрипте строка 'agnea1' является именем пользователя на машине с Linux, на которую производится резервирование данных.

```
#!/bin/bash
```

```
clear
echo Initialising ...
checkdate=`date | awk '{print $1}'

if [ -f "~agneal/backup-dir/backup-data" ]; then

    echo "ERROR: No config file for today!"
    echo "FATAL!"
    exit 1
fi

if [ -d "~agneal/backup-dir/temp" ]; then

    echo "ERROR: No tempoary directory found!"
    echo
    echo "Attempting to create"
    cd ~agneal
    cd backup-dir
    mkdir temp
    echo "Directory Made - temp"
fi

if [ "$1" = "" ]; then

    echo "ERROR: enter in a machine name (ie: cdwriter)"
    exit 1
fi

if [ "$2" = "" ]; then

    echo "ERROR: enter in a SMB (Lan Manager) Resource (ie: work)"
    exit 1
fi

if [ "$3" = "" ]; then

    echo "ERROR: enter in an IP address for $1 (ie:
    130.xxx.xxx.52)" exit 1
fi

#####
# Main Section
#
#####

cd ~agneal/backup-dir/temp
rm -r ~agneal/backup-dir/temp/*
cd ~agneal/backup-dir/

case "$checkdate"
in
    Mon)
        echo "Backing up for Monday"
        cat backup-data | /usr/local/samba/bin/smbclient
        \\$1\\$2 -I$3 -N echo "Complete"
```

```
        if [ -d "~agneal/backup-dir/Monday" ]; then
            echo "Directory Monday Not found ..."
            making" mkdir
                ~agneal/backup-dir/Monday
        fi

    echo "Archiving ..."
    cd ~agneal/backup-dir/temp
    tar -cf monday.tar *                echo "done ..."
    rm ~agneal/backup-dir/Monday/monday.tar
    mv monday.tar ~agneal/backup-dir/Monday
;;

Tue)

echo "Backuping for Tuesday"
cat backup-data | /usr/local/samba/bin/smbclient
\\\\$1\\$2 -I$3 -N echo "Complete"

        if [ -d "~agneal/backup-dir/Tuesday" ]; then
            echo "Directory Tuesday Not found ..."
            making" mkdir
                ~agneal/backup-dir/Tuesday
        fi

    echo "Archiving ..."
    cd ~agneal/backup-dir/temp
    tar -cf tuesday.tar *
    echo "done ..."
    rm ~agneal/backup-dir/Tuesday/tuesday.tar
    mv tuesday.tar ~agneal/backup-dir/Tuesday
;;

Wed)

echo "Backuping for Wednesday"
cat backup-data | /usr/local/samba/bin/smbclient
\\\\$1\\$2 -I$3 -N echo "Complete"

        if [ -d "~agneal/backup-dir/Wednesday" ]; then
            echo "Directory Wednesday Not found
            ... making" mkdir
                ~agneal/backup-dir/Wednesday
        fi

    echo "Archiving ..."
    cd ~agneal/backup-dir/temp
    tar -cf wednesday.tar *
    echo "done ..."
rm ~agneal/backup-dir/Wednesday/wednesday.tar
mv wednesday.tar ~agneal/backup-dir/Wednesday
;;

Thu)

echo "Backuping for Thrusday"
cat backup-data | /usr/local/samba/bin/smbclient
\\\\$1\\$2 -I$3 -N echo "Complete"

        if [ -d "~agneal/backup-dir/Thursday" ]; then
            echo "Directory Thrusday Not found ..."
            making" mkdir
```

```

                                ~agneal/backup-dir/Thursday
        fi
        echo "Archiving ..."
        cd ~agneal/backup-dir/temp
        tar -cf thursday.tar *
        echo "done ..."
        rm ~agneal/backup-dir/Thursday/thursday.tar
        mv thursday.tar ~agneal/backup-dir/Thursday
        ;;

    Fri)

        echo "Backing up for Friday"
        cat backup-data | /usr/local/samba/bin/smbclient
        \\$1\\$2 -I$3 -N echo "Complete"

                if [ -d "~agneal/backup-dir/Friday" ]; then
                        echo "Directory Friday Not found ...
                        making" mkdir
                                ~agneal/backup-dir/Friday
                fi

        echo "Archiving ..."
        cd ~agneal/backup-dir/temp
        tar -cf friday.tar *
        echo "done ..."
        rm ~agneal/backup-dir/Friday/friday.tar
        mv friday.tar ~agneal/backup-dir/Friday
        ;;

    *)

        echo "FATAL ERROR: Unknown variable passed for day"
        exit 1;;

esac
#####

```

Вот скрипт предоставленный Dan:

```

#!/bin/bash

BACKDIR=3D/backup
WINCMD=3D/usr/sbin/smbclient

function CopyWinHost(){

# tars and gzips "windows shares" to a local directory using samba's
# smbclient
# argument 1 is the remote host window's host name
# argument 2 is the share name to be backed up

        echo $1,$2,$3
        REMOTE=3D$1
        SHARE=3D$2
        DEST=3D$3

# create a tarred gzip file using samba to copy direct from a
# windows pc
# 12345 is a password. Needs some password even if not defined on
# remote system

```

```

$WINCMD \\\\$REMOTE\\$SHARE 12345 -Tc -|gzip > $DEST
echo `date`: Done backing up "$REMOTE" to "$DEST
echo
}

function CopyUnixHost(){

# tars and gzips a directory using rsh
# argument 1 is the name of the remote source host
# argument 2 is the full path to the remote source directory
# argument 3 is the name of the local tar-gzip file.  day of week
# plus .tgz will be appended to argument 3

REMOTE=3D$1
SRC=3D$2
DEST=3D$3

if rsh $REMOTE tar -cf - $SRC |gzip > $DEST; then
    echo `date`: Done backing up "$REMOTE":"$SRC" to "$DEST
else
    echo `date`: Error backing up "$REMOTE":"$SRC" to "$DEST
fi
}

# $1: win=3Dbackup windows machine, unix=3Dbackup unix machine
case $1 in
win)
    # $2=3D remote windows name, $3=3Dremote share name,
    # $4=3Dlocal destination directory
    CopyWinHost $2 $3 $4;;
unix)
    # $2 =3D remote host, $3 =3D remote directory,
    # $4 =3D destination name
    CopyUnixHost $2 $3 $4;;
esac

```

12 Использование Samba между маршрутизируемыми сетями

Andrew Tridgell заявил, что просмотр машин с SMB через маршрутизаторы является проблематичной задачей. Вот его предложения для исправления:

----- — Для просмотра пересекающихся подсетей (например маршрутизируемых) вы должны сделать следующее. Существуют другие способы, но они более сложные и склонны к появлению ошибок:

- 1) все компьютеры, которые вы хотите видеть должны использовать один сервер WINS (это могут делать Samba или NT)
- 2) основной броузер для каждой сети должен быть либо NT, либо Samba. (Win9X неправильно распространяет межсетевую информацию)
- 3) Вы должны использовать одно имя рабочей группы во всех подсетях. Это не обязательно, но это может быть простым способом достижения успеха. Если вы не можете достичь этого, то вы должны организовать распространение информации между подсетями. (Это не распространение через WINS). Информация распространяется двумя способами: i) каждый основной броузер извещает анонсы рабочих групп от других основных броузеров в своем широковещательном домене ii) каждый не-Win9X основной броузер связывается с глобальным DMB для рабочей группы (обычно это контролер домена или машина с Samba, отмеченная как мастер домена) и периодически

обменивается полной информацией. ————— —

Также, Rakesh Bharania указал, что маршрутизатор Cisco может быть настроен для передачи данных SMB способом, который позволит просмотр серверов. Он предлагает настроить интерфейс маршрутизатора, который обслуживает SMB-машины с помощью следующей команды:

```
ip helper-address x.x.x.x
```

где x.x.x.x это IP-адрес SMB-сервера.

13 Авторские права

Авторскими правами на этот документ владеет David Wood, 1996-9. Он может воспроизводиться в любой форме и свободно распространяться, при сохранении файла неизменным, включая это заявление.

14 Благодарности Специальные благодарности Andrew Tridgell (*tridge@linuxcare.com*) за начало и направление проекта Samba и за сохранение этого документа в подлиннике.

Brad Marshall (*bmarshall@plugged.net.au*) и Jason Parker (*jparker@plugged.net.au*) нашедших время, проявивших настойчивость, написавших скрипт и проводивших эксперименты.

Adam Neat (*adamneat@ipax.com.au*) и Dan Tager (*dtager@marsala.com*), предоставившим скрипт для сохранения данных с машин с Windows на машину с Linux.

Matthew Flint (*matthew@philtrum.demon.co.uk*) рассказавшему мне об использовании настройки 'interfaces' в файле smb.conf.

Oleg L. Machulskiy (*machulsk@shade.msu.ru*), Jeff Stern (*jstern@eclectic.ss.uci.edu*), Dr. Michael Langner (*langner@fiz-chemie.de* and Erik Ratcliffe (*erik@caldera.com*) предложивших изменения в раздел об организации доступа к принтерам Linux для машин с Windows.

Alberto Menegazzi (*flash.egon@iol.it*) предоставившему мне настройки для MagicFilter, чтобы он могу обеспечивать доступ к принтерам Windows для машин с Linux.

Rakesh Bharania (*rbharani@cisco.com*) за данные по настройке маршрутизатора Cisco.

Rich Gregory (*rtg2t@virginia.edu*) и других, предоставивших данные о пакете smbfs и его использовании.

Andrea Girotto (*icarus@inca.dei.unipd.it*) пославшему мне некоторое количество пожеланий относительно данного документа.

Также благодарю всех национальных переводчиков, которые перенесли этот документ в мир не говорящий по Английски: Takeo Nakano (*nakano@apm.seikei.ac.jp*), Klaus-Dieter Schumacher (*Klaus-Dieter.Schumacher@fernuni-hagen.de*), Andrea Girotto (*icarus@inca.dei.unipd.it*), Mathieu Arnold (*arn_mat@club-internet.fr*), Stein Oddvar Rasmussen (*Stein@kongsberg-energi.no*) Nilo Menezes (*nmenezes@n3.com.br*) и многих других. о которых я не имею контактной информации.