

**Сервера IP телефонии
PACS-E1, Asteroid2-MV16**

**Конфигурация и управление
Версия 1.0
02.04.2010**

Разработчик и производитель: ООО «Парабел»
630090, Новосибирск-90, а/я 126
<http://www.parabel.ru>
Email: info@parabel.ru
Тел/факс: +7-383-2138707

Содержание

1. ВВЕДЕНИЕ	6
2. ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ	7
3. ОСОБЕННОСТИ ФАЙЛОВОЙ СИСТЕМЫ	8
4. КОНФИГУРАЦИЯ LINUX В ЦЕЛОМ	10
5. БАЗОВЫЕ НАВЫКИ	11
5.1. Как изменить IP адрес	11
5.2. Как изменить имя системы.....	11
5.3. ИЗМЕНЕНИЕ СИСТЕМНОГО ВРЕМЕНИ И ДАТЫ	11
5.4. Как поменять пароль.....	12
5.5. Как добавить нового пользователя	12
5.6. Как сохранить текущую конфигурацию	12
5.7. Как перезагрузить или выключить систему	13
5.8. Как узнать конфигурацию сетевых интерфейсов.....	13
5.9. Как посмотреть таблицу маршрутизации.....	13
5.10. Как посмотреть загрузку системы	13
5.11. Как посмотреть список процессов	13
5.12. Как удалить процесс	14
5.13. Где хранятся сообщения системы	14
5.14. Как настроить доступ к файлам WINDOWS.....	14
5.15. РАБОТА С ПАКЕТАМИ DEBIAN.....	15
5.16. Список основных предустановленных пакетов DEBIAN	15
6. КОНФИГУРАЦИЯ IP ИНТЕРФЕЙСОВ (IFCONFIG)	18
7. КОНФИГУРАЦИЯ VLAN ИНТЕРФЕЙСОВ (VCONFIG)	19
8. УПРАВЛЕНИЕ МОСТОМ (BRCTL)	20
9. СТАТИЧЕСКАЯ МАРШРУТИЗАЦИЯ (ROUTE)	22
10. СЕТЕВАЯ СТАТИСТИКА (NETSTAT)	23
11. ДИНАМИЧЕСКАЯ МАРШРУТИЗАЦИЯ	24
12. ФИЛЬТРАЦИЯ И NAT (IPTABLES)	25
12.1. Команды	26
12.2. Задание правил обнаружения пакетов (поле RULE).....	26
12.3. Действия при обнаружении пакета (-J опция)	27
12.4. Дополнительные возможности IPTABLES	27
13. УПРАВЛЕНИЕ ТРАФИКОМ (TC)	28
14. КОНФИГУРАЦИЯ E1 ПОДСИСТЕМЫ	29
14.1. Физический уровень E1	29
14.2. Канальные интервалы для передачи данных и сигнализации	30
14.3. Закон кодирования.....	30
14.4. РЕЖИМ ЛОКАЛИЗАЦИИ.....	31
14.5. Коммутация каналов	32
14.6. Утилита esfg	33
14.7. Сигнализация ISDN PRI	34
14.7.1. Конфигурация	34
14.7.2. Диагностика протокола.....	35
14.8. Сигнализация SS7.....	36
15. КОНФИГУРАЦИЯ FXO/FXS КАНАЛОВ	37
16. ВОЗМОЖНОСТИ IP ATC	39
16.1. ВВЕДЕНИЕ	39
16.2. КОНФИГУРАЦИОННЫЕ ФАЙЛЫ ASTERISK.....	39
16.3. Источники информации по IP ATC ASTERISK	41

ПРИЛОЖЕНИЕ 1. ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА.....42

1. Введение

В данном документе приводится описание программного обеспечения (ПО) серверов PACS-E1, Asteroid2-MV16, рассматриваются особенности управления и конфигурации. Программное обеспечение представляет собой пакет GNU/Linux с возможностью автономного старта и загрузки с USB flash памяти.

ПО сервера поддерживает следующие протоколы и сервисы:

VOIP сервис

Базируется на Asterisk v1.6

Кодеки G.711 (a,m), G.729AB, GSM

Протоколы SIP, H.323, RTP

Маршрутизация

BGP4, BGP4+, OSPFv2, OSPFv3, RIPv1, RIPv2, RIPng

Аутентификация пользователей

RADIUS

Безопасность

IP firewall, NAT

Статистика

IP accounting

Управление и конфигурация

Локальная консоль (SVGA, Kbd)

ssh, telnet, ftp, nfs

Диагностика

traceroute, dig, tcpdump, netcat

С помощью менеджера пакетов дополнительно может быть установлена любая утилита из стандартного репозитория Debian.

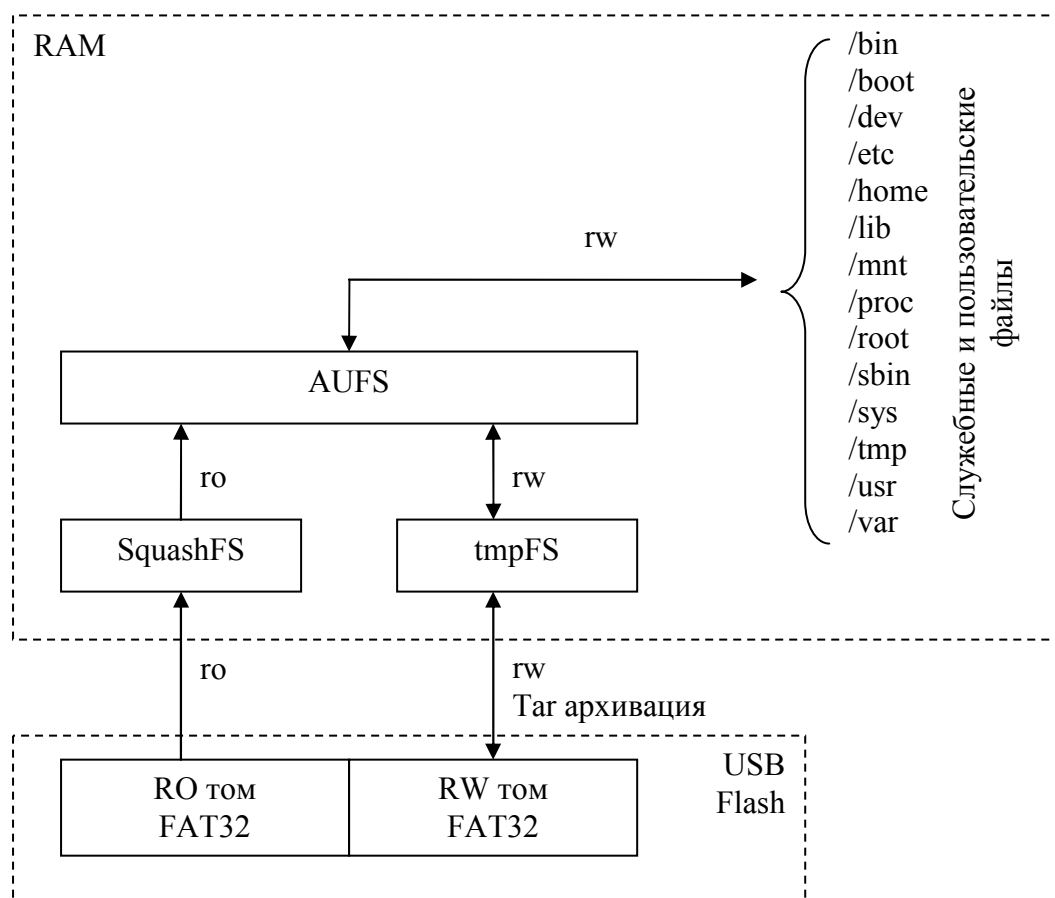
2. Лицензионное соглашение

Рассмотренное в данном документе программное обеспечение не является предметом продажи или каких-либо иных сделок с коммерческой направленностью, а поставляется производителем бесплатно. Стоимость ПО не заложена в стоимость оборудования. Производитель не налагает авторских прав на данное ПО и допускает его дополнение пользователями.

Пакет ПО собран из открытых источников в Интернете и поставляется «как есть». Производитель не несет ответственности за возможные ошибки или несоответствия документации, допущенные разработчиками открытого кода. Тем не менее, производитель налагает на себя обязанность по технической поддержке пользователей изделий PACS-E1 и Asteroid2-MV16 в порядке консультативной помощи. Производитель также гарантирует совместимость поставляемого ПО с аппаратурой.

3. Особенности файловой системы

Структура файловой системы сервера PACS-E1 (Asteroid2-MV16) изображена на рисунке. После загрузки, в рабочем режиме ПО сервера работает **только** в оперативной памяти, без обращения к flash диску. Как видно из рисунка, рабочая файловая система состоит из двух частей – только для чтения (ro) и только для записи (rw).



Ro часть содержит ядро Linux, системные утилиты и заводскую конфигурацию. Rw часть содержит все изменения, внесенные пользователем во время работы. Файловые системы ro и rw объединяются в одно целое с помощью файловой системы AUFS, которая предоставляет для пользователя стандартную корневую файловую систему Linux, доступную для записи. Таким образом, отслеживание изменений и записей в файлы происходит незаметно для пользователя. Во время работы сервера содержимое ro и rw файловых систем можно посмотреть в директориях `/.pb/ro/` и `/.pb/rw/` соответственно.

Инициализация обеих частей файловой системы происходит из разных томов USB flash носителя. Так, для ro части используется том размером около 200 Мб с файловой

системой FAT32. Здесь же содержится MBR и загрузчик Grub. Для gw системы отводится весь оставшийся объем flash носителя, который также организован в формате FAT32. Содержимое gw системы записывается в этот том в виде TAR архивов. Запись содержимого gw системы инициируется только пользователем и происходит в трех случаях – при перезапуске системы (командой reboot), при останове системы (команда halt) и при записи конфигурации во flash (команда writeflash).

При старте системы происходит обратный процесс – образ gw системы распаковывается из tar архивов, расположенных в RW томе flash диска и записывается в оперативную память сервера.

Рассмотренное строение файловой системы обеспечивает удобство конфигурации сервера, а также обеспечивает высокую отказоустойчивость.

4. Конфигурация Linux в целом

Изменить конфигурацию сервера можно двумя способами – оперативно и permanently, через конфигурационные файлы. Оперативные изменения конфигурации происходят с помощью консольных команд и наступают сразу, без перезагрузки сервера. Например, изменить IP адрес можно консольной командой **ifconfig**:

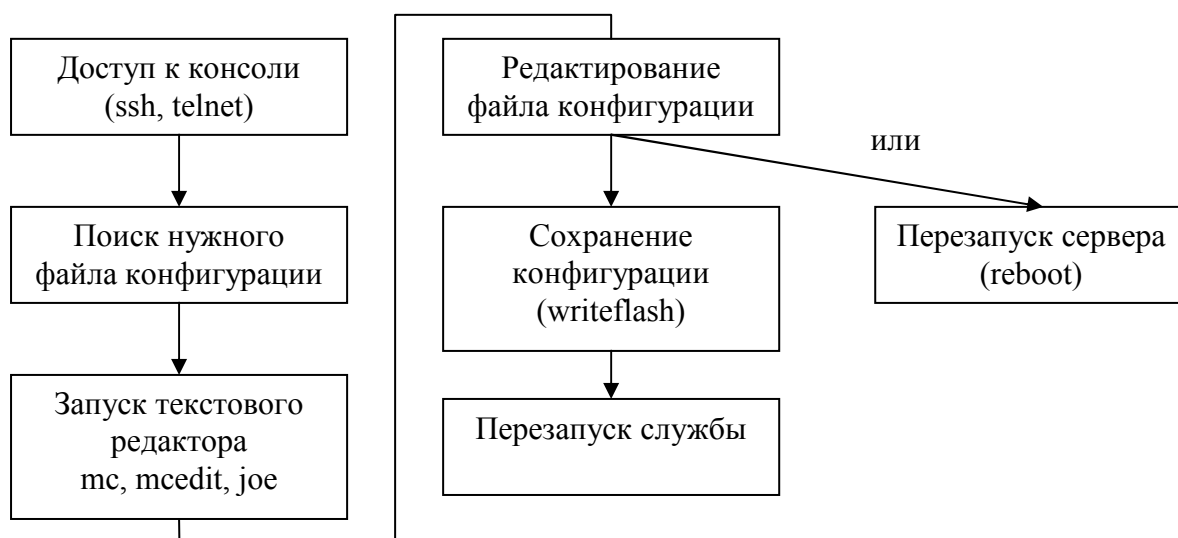
```
# ifconfig eth0 192.168.1.1
```

Адрес **192.168.1.1** присвоится интерфейсу **eth0** сразу же. Оперативные изменения конфигурации будут действительны до первой перезагрузки системы. Их невозможно сохранить во flash память.

Для permanentного изменения конфигурации необходимо редактировать конфигурационные файлы. Конфигурационные файлы в большинстве случаев имеют текстовый формат и редактируются встроенными редакторами, входящими в состав ПО сервера. Например, изменение IP адреса может быть сделано путем редактирования файла **/etc/network/interfaces**. Сделать это можно с помощью файлового менеджера **mc** или редактора **joe**. После изменения конфигурационного файла, конфигурацию системы в целом нужно сохранить командой **writelflash**. Чтобы новые параметры стали актуальными, необходимо или перезапустить весь сервер или соответствующий сервис. В нашем случае достаточно перезапустить сетевую подсистему:

```
# /etc/init.d/networking restart
```

Таким образом, при изменении конфигурации, действия пользователя укладываются в следующую диаграмму:



5. Базовые навыки

5.1. Как изменить IP адрес

При загрузке ОС, параметры для сетевых интерфейсов берутся из файла “/etc/network/interfaces”.

- “auto lo eth0 eth1” указывает ОС, что надо сконфигурировать интерфейсы eth0, eth1.
- “iface eth0 inet dhcp” указывает, что параметры интерфейса eth0 надо получить по протоколу dhcp.
- Следующие строки описывают интерфейс eth1 с статическим адресом.

```
iface eth1 inet static
address 192.168.1.1
netmask 255.255.0.0
name Private network
```

При загрузке, ОС интерпретирует файл “/etc/network/interfaces” и вызывает команду “*ifconfig*” с указанными параметрами. Изменить адрес интерфейса без перезагрузки и сохранения можно командой “*ifconfig*”.

Пример:

```
“ifconfig eth1 192.168.1.1 netmask 255.255.255.0”
```

установит интерфейсу eth1 адрес 192.168.1.1 с маской сети 255.255.255.0 .

5.2. Как изменить имя системы

Имя системы хранится в файле /etc/hostname , и считывается операционной системой (ОС) единожды (при загрузке). В зависимости от настройки, сервисы (демоны) могут использовать это имя, или хранить свое имя.

5.3. Изменение системного времени и даты

Сервер имеет часы реального времени (RTC), работающие даже при выключенном питании. Настройка часов реального времени осуществляется командой “*hwclock*”.

Запустившись, операционная система берет время RTC, добавляет к нему (если необходимо) смещение временной зоны, и в дальнейшем обслуживает время самостоятельно. Управление временем ОС занимается команда “*date*”.

Однако часы сервера имеют конечную точность, и могут уходить. Чтобы это избежать, рекомендуется настроить сервис времени (*openntpd*). Сервис *openntpd* подключается к мировым серверам времени, и плавно подстраивает локальные часы (шаг подстройки примерно 0.1 сек за 60 секунд).

Настройка *openntpd* осуществляется в файле */etc/openntpd/ntpd.conf*.

5.4. Как поменять пароль

Поменять пароль пользователя *user1* можно командой “*passwd user1*”. **Обязательно смените пароль пользователя *root* и сохраните конфигурацию, см. 5.6. Как сохранить текущую конфигурацию !**

5.5. Как добавить нового пользователя

Добавить нового пользователя можно командой *adduser*.

Пример (добавить пользователя *user1*):

“*adduser user1*”

5.6. Как сохранить текущую конфигурацию

Сохранить директорию */etc/* можно принудительно командой “*writeflash*”.

Для сохранения всех измененных данных, например установленные пакеты и пользовательские данные, необходимо перезагрузить или выключить систему. Системные “скрипты” сохранят все измененные данные в архив.

5.7. Как перезагрузить или выключить систему

Перезагрузить систему можно командой `“reboot”`.

Выключить систему можно командами `“shutdown -h now”`, `“halt”`.

5.8. Как узнать конфигурацию сетевых интерфейсов

Конфигурацию сетевых интерфейсов можно запросить командой `“ifconfig”`.

Конфигурацию выбранного интерфейса можно запросить, добавив имя интерфейса параметром команды `“ifconfig”`.

Команда `“ifconfig”` показывает только настроенные интерфейсы (статус UP).

Список всех интерфейсов и “сырую” статистику можно запросить командой `“cat /proc/net/dev”`

5.9. Как посмотреть таблицу маршрутизации

Текущую таблицу маршрутизации можно посмотреть командой `“route”`.

5.10. Как посмотреть загрузку системы

Текущую загрузку системы можно посмотреть командами `“top”` и `“htop”`.

Поля имеют следующие значения (проценты, относительно всех процессоров):

- us = user, загрузка процессоров непосредственно приложениями.
- sy = system, загрузка процессоров ядром операционной системы и драйверами.
- id = idle, простой процессоров.

Команда `“uptime”` в поле `“load average”` отобразит среднюю загрузку за 1, 5, 15 минут.

5.11. Как посмотреть список процессов

Список можно запросить командой `ps`.

Также можно отфильтровать процессы командой `“grep”` или пролистать их командой `“less”`.

Пример:

`“ps auxf | grep init”` отобразит все строки выданные `“ps”` отфильтровав их по строке `“init”`.

5.12. Как удалить процесс

Удалить процесс можно командой “*kill* PID”, где PID – идентификатор процесса. Убить процесс (завершить аварийно) можно добавив параметр -9 к команде. “*kill -9*” рекомендуется использовать если “*kill*” без параметра не удалила процесс.

Удалить все процессы по имени можно командой *killall* (используйте с осторожностью).

5.13. Где хранятся сообщения системы

Типично все сообщения служб и системы хранятся в директории */var/log/*. Сообщения ядра ОС попадают в файл *kern.log*, сообщения загрузки системы в файл *dmesg*. Сообщения ядра и основных служб попадают в файл *messages*.

Посмотреть сообщения можно командой “*less*”.

Смотреть поступающие сообщения можно командой “*tail -f*”.

Примеры:

“*less /var/log/messages*” покажет сообщения в файле “*messages*”

“*tail -f /var/log/messages*” покажет все поступающие сообщения ядра и базовых служб.

5.14. Как настроить доступ к файлам Windows

Подключить сетевой диск Windows можно командой *mount*.

“*mount -t smbfs -o username=user1 //192.168.1.2/folder1 /mnt/folder1/*”

подключит сетевую директорию *folder1* с компьютера IP=192.168.1.2 в директорию */mnt/folder1* используя имя пользователя *user1* и пароль, который пользователь введет по запросу команды *mount*.

Получить доступ к файлам сервера можно по протоколу SSH (SFTP/SCP). Для этого можно использовать программу *winscp*, или установить дополнение *winscp* для *far* или *total commander*. Подключаться к серверу нужно используя имеющиеся учетные записи.

5.15. Работа с пакетами Debian

Пакетами Debian управляет программа dpkg.

Ее задача отображать, устанавливать и удалять пакеты, а также отслеживать зависимости пакетов. Поверх программы работает программа apt. Ее задача устанавливать пакеты из источников (CD, Internet,...), проверять зависимости через dpkg и устанавливать все пакеты, необходимые для удовлетворения зависимостей.

Примеры:

- “apt-cache search | grep asterisk” отобразит все доступные пакеты, содержащие слово “asterisk”.
- “dpkg -l | grep asterisk” отобразит все установленные пакеты, содержащие слово “asterisk”.
- “apt-get install asterisk” установит все пакеты, и все пакеты необходимые для пакета “asterisk”.

Список источников пакетов содержится в файле “/etc/apt/sources.list”.

5.16. Список основных предустановленных пакетов Debian

Acl	
Apache	Web сервер Apache2
Asterisk	IP АТС Asterisk
dahdi	Драйвер для периферийного оборудования Asterisk
Fuse	Драйвер файловой системы FUSE
Htop	Отображение загрузки ОС и памяти процессами
Ifenslave	Объединение сетевых интерфейсов (bonding)
Iotop	Отображение файловой загрузки процессами
Ircad	Сбор статистики соединений
Joe	Редактор
Mc	Файловый командер
minicom	Терминал
netcat	Утилита для создания трубы через tcp,udp. Для отладочных целей.
nfs-kernel-server	NFS сервер
openntpd	Сервер и клиент времени
openssh	Ssh сервер и клиент
openswan	Ipssec сервер и клиент
openvpn	VPN сервер поверх TCP,UDP
perl	Интерпретатор perl5
php5	Интерпретатор PHP5
pppd	Сервер и клиент PPP
pppoed	PPP через Ethernet (PPPoE)

quagga	Сервера динамической маршрутизации rip, ripng, ospf, bgp
samba	Сервер и клиент SMB
tcpdump	Sniffer
Top	Отображение загрузки ОС и памяти процессами

6. Конфигурация IP интерфейсов (ifconfig)

Назначение и удаление IP адресов на интерфейсах передачи данных осуществляется командой **ifconfig**.

ifconfig без параметров выводит список интерфейсов с указанием IP адресов, масок и ряда других параметров, а также выводит статистику пакетов по интерфейсам.

Для назначения IP адресов используется следующий синтаксис:

ifconfig <interface> [address] [options]

где,

interface – имя интерфейса (eth0, hdlc1 и т.д.)

address – IP адрес интерфейса (например, 100.0.0.1)

Опции:

[netmask <address>] – установить IP маску

[broadcast <address>] – установить широковещательный адрес интерфейса

[pointopoint <address>] – установить адрес партнера для соединений точка-точка

[up | down] – включить/выключить интерфейс

ПРИМЕР:

```
ifconfig eth0 192.168.1.1 netmask 255.255.255.0
```

Подробная информация по конфигурации IP интерфейсов приведена [здесь](#).

7. Конфигурация VLAN интерфейсов (vconfig)

Команда **vconfig** используется для конфигурации vlan (IEEE802.1Q) интерфейсов. Добавление виртуальных интерфейсов возможно только для интерфейса Ethernet.

Использование:

add <ifname> <vlan_id> - добавить виртуальный интерфейс с идентификатором **vlan_id** на реальный интерфейс **ifname**. Имя виртуального интерфейса будет выглядеть как **eth0.vlan_id**, где **vlan_id** – десятичное число 0..4095.

rem <vlan_name> - удалить виртуальный интерфейс с именем **vlan_name**.

Пример:

```
vconfig add eth0 45
```

```
ifconfig eth0.45 192.168.45.1 netmask 255.255.255.0
```

8. Управление мостом (brctl)

Команда **brctl** используется для добавления и удаления моста (bridge) в систему, назначения интерфейсов, подключенных к мосту и мониторинга работы моста. Интерфейсы передачи данных, подключенные к мосту, образуют группу моста и напрямую недоступны для маршрутизации. В пределах этой группы между интерфейсами происходит передача данных по протоколам моста. Маршрутизация может использоваться только между группой как целым и остальными интерфейсами, не включенными в группу моста. Группе интерфейсов, включенных в мост, приписывается виртуальный интерфейс передачи данных, которому может быть назначен свой IP адрес и маска.

Подробная информация по работе моста и протокола STP приведена [здесь](#).

Использование:

brctl <command> [parameters]

команды могут быть следующими:

addbr <brname> – добавить новый мост в систему с именем **brname**. Имя указывается администратором и может быть произвольным, например, **br0**, **br1** и т.д. Далее указанное имя может использоваться для присвоения IP адреса и маски командой **ifconfig**.

delbr <brname> - удалить мост с указанным именем из системы

addif <brname> <ifname> - добавить в группу моста с именем **brname** интерфейс **ifname**. После этого интерфейс становится недоступным для маршрутизации.

delif <brname> <ifname> - удалить интерфейс **ifname** из группы моста **brname**.

brctl stp <on/off> - включить (**on**) / выключить (**off**) поддержку протокола Spanning Tree. Протокол STP предотвращает образование петель и обеспечивает выбор наилучшей топологии сети.

brctl show – вывести на консоль список мостов, существующих в системе

brctl showmacs <brname> - вывести на консоль список MAC адресов, обнаруженных в сегменте моста **brname**.

brctl showstp <brname> - вывести статистику протокола STP для моста **brname**.

После добавления нового моста в систему и подключения к нему интерфейсов передачи данных, мосту может быть назначен IP адрес командой **ifconfig**.

Примечание:

Мост может включать интерфейсы “выглядящие” как Ethernet (у них должны быть 6-байтные адреса отправителя и получателя) и интерфейсы должны иметь одинаковые MTU.

Для hdlc интерфейсов, на канальном уровне могут использоваться протоколы hdlc-eth, cisco-bridge (см. “**Ошибка! Источник ссылки не найден.**”).

Также можно включать в мост виртуальные интерфейсы, порожденные туннелями, например vtund, openvpn, ...

Пример конфигурации моста и присвоения ему IP адреса:

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 hdlc0
brctl stp on
ifconfig br0 192.168.1.54 netmask 255.255.255.0
```

9. Статическая маршрутизация (route)

С помощью команды **route** можно вручную добавлять и удалять маршруты из таблицы маршрутизации системы. **route** без параметров выводит известные системе маршруты.

ИСПОЛЬЗОВАНИЕ:

add [-net | -host] IP [netmask NM] [gw GW] [metric N] [dev IF] – добавить маршрут

del [-net | -host] IP [netmask NM] [gw GW] [metric N] [dev IF] – удалить маршрут

где,

IP – ip адрес сети или хоста, на которые указывает маршрут. В случае указания маршрута на сеть необходимо задать параметр **netmask**.

NM – сетевая маска, например 255.255.255.0.

Для указания маршрута «по умолчанию», вместо **IP** и **NM** можно использовать ключевое слово **default**.

GW – ip адрес сервера (если необходимо)

metric N – метрика маршрута, где N – десятичное число 0..15. Метрика используется сервисом динамической маршрутизации и должна соответствовать числу промежуточных IP серверов до указанной подсети (хоста). Сети, доступные из маршрутизатора напрямую, должны иметь метрику 0, доступные через два сервера – метрику 2.

IF – имя интерфейса, через который должны отправляться пакеты (eth0, hdlc0 и т.д.).

Примеры:

```
route add default gw 100.0.0.1
```

```
route add -net 192.168.1.0 netmask 255.255.255.0 gw 100.0.0.1
```

Подробная информация по таблице маршрутизации и использованию команды **route** приведена [здесь](#) или [здесь](#).

10. Сетевая статистика (**netstat**)

Команда **netstat** полезна для получения информации о состоянии сетевой системы Linux. Она позволяет выводить информацию о маршрутах, сетевых интерфейсах и соединениях.

Описание команды приводится [здесь](#).

11. Динамическая маршрутизация

Динамическая маршрутизация в сервере PACS-E1 реализована на пакете Quagga.

Quagga - пакет программ, реализующих протоколы маршрутизации, основанных на TCP/IP и поддерживает такие протоколы как RIPv1, RIPv2, RIPv6, OSPFv2, OSPFv3, BGP-4, и BGP-4+. Quagga также поддерживает BGP Route Reflector и Route Server behavior. В дополнение к традиционному протоколу IPv4 Quagga также поддерживает протоколы маршрутизации для IPv6.

Подробное описание пакета и его конфигурация описаны [здесь](#).

По умолчанию, демоны динамической маршрутизации не запущены. Примеры конфигурационных файлов по каждому протоколу маршрутизации можно найти в директории `/usr/share/doc/quagga/examples/`.

12. Фильтрация и NAT (iptables)

Управление встроенным фильтром ОС Linux осуществляется командой **iptables**. Конфигурирование фильтра сводится к заданию правил распознавания пакетов и задания действий, в случае обнаружения нужного пакета. Правила разбиты на цепочки. Цепочки правил разбиты на основные группы – таблицы. Наиболее часто используемыми таблицами являются **nat** и **filter**. При манипуляции с цепочками правил администратор должен указывать таблицу, которой они принадлежат.

В каждой таблице существует predetermined набор цепочек:

Таблица **filter**:

INPUT - все пакеты, идущие непосредственно на IP адрес маршрутизатора

FORWARD – все маршрутизируемые пакеты

OUTPUT – все пакеты, созданные и отправленные самим маршрутизатором

Таблица **nat**:

PREROUTING – пакеты до маршрутизации

OUTPUT - все пакеты, созданные и отправленные самим маршрутизатором

POSTROUTING – пакеты после маршрутизации

Использование iptables:

iptables –[AD] chain rule [options]

iptables –I chain [rulenum] rule [options]

iptables –R chain rulenum rule [options]

iptables –D chain rulenum [options]

iptables –[LFZ] [chain] [options]

iptables –N chain

iptables –X chain

iptables –P chain target [options]

где,

chain – имя цепочки

rule – спецификация правила поиска пакета, см. задание правил

rulenum – номер правила в цепочке

target – с ключом **–P** означает действие по умолчанию (политика цепочки по умолчанию)

12.1. Команды

- N** – создать новую цепочку
- X** – удалить цепочку
- A** – добавить правило к выбранной цепочке
- D** – удалить правило из цепочки (указать номер правила или его спецификацию)
- R** – заменить правило с заданным номером
- I** – вставить правило перед правилом с номером *rule*
- L** – вывести список всех правил указанной цепочки
- F** – стереть все правила в цепочке (если цепочка не указана, стереть все)
- Z** – стереть счетчики пакетов во всех правилах цепочки

12.2. Задание правил обнаружения пакетов (поле *rule*)

- p protocol** – протокол (**tcp**, **udp**, **icmp**, **all**)
- s addr/[mask]** – IP адрес отправителя пакета
- d addr/[mask]** – IP адрес получателя пакета

Для протокола **tcp** допускаются также условия:

- source-port port[:port]** – порт (или диапазон портов) отправителя **tcp**
- destination-port port[:port]** – порт (или диапазон портов) получателя **tcp**
- syn** – обнаруживать все пакеты с установленным флагом **SYN** и сброшенными

флагами **ACK**, **FIN** (**tcp** пакеты, инициирующие соединение)

Для протокола **udp** допускаются условия:

- source-port port[:port]** – порт (или диапазон портов) отправителя **udp**
- destination-port port[:port]** – порт (или диапазон портов) получателя **udp**

Большинство полей допускает задание с инверсией (символ «!»). Например, **-p !tcp** означает все протоколы, кроме **tcp**.

12.3. Действия при обнаружении пакета (-j опция)

-j target – действие, которое необходимо выполнять при обнаружении пакета. Поле target может означать имя другой цепочки или одно из predefined действий.

Для таблицы **filter** действия задаются ключевыми словами:

ACCEPT - пропустить пакет

DROP - выбросить пакет

RETURN - прекратить просматривать текущую цепочку и вернуться в предыдущую

Для таблицы **nat**, цепочка **POSTROUTING**:

SNAT – преобразовать адрес отправителя, дополнительно указать опцию **--to-source**

--to-source ipaddr[-ipaddr][:port-port] – адрес отправителя после преобразования будет выбран из диапазона **ipaddr-ipaddr**. Если протокол tcp или udp, дополнительно можно указать диапазон портов отправителя.

Для таблицы **nat**, цепочки **PREROUTING** и **OUTPUT**:

DNAT – преобразовать адрес получателя, дополнительно указать опцию **--to-destination**

--to-destination ipaddr[-ipaddr][:port-port] – адрес получателя после преобразования будет выбран из диапазона **ipaddr-ipaddr**. Если протокол tcp или udp, дополнительно можно указать диапазон портов получателя.

12.4. Дополнительные возможности iptables

В данном описании возможности iptables описаны не полностью, для более полной справки обращаться к [документации](#).

13. Управление трафиком (tc)

Сервер PACS-E1 предоставляет широкие возможности по управлению пропускной способностью для различных видов трафика. Так, IP пакеты могут быть классифицированы по различным признакам и в зависимости от этого помещены в очереди на отправку с разными приоритетами и разной пропускной способностью. Таким образом, клиентам, подключенным к серверу, может быть обеспечен сбалансированный доступ к каналам передачи данных. Управление трафиком осуществляется с помощью утилиты **tc**. Вопросы управления пропускной способностью подробно описаны в [документе](#).

14. Конфигурация E1 подсистемы

14.1. Физический уровень E1

Все E1 порты в системе должны быть описаны ключевым словом **span**.

Файл с конфигурацией	Параметр
/etc/dahdi/system.conf	span = <span_num>,<timing>,<LBO>,<framing>,<coding>[,crc4]

где

span_num – номер порта E1 (от 1 до максимального номера порта в плате)

timing – использовать ли порт как источник синхронизации

0 – порт адаптера ведущий по E1

1 и более – порт ведомый по E1 и является одним из источников синхронизации адаптера.

LBO – параметр не используется, ставить 0.

Framing – тип телефонной сигнализации, ставить ccs или cas.

Coding – кодирование в линии, может принимать значения amī или hdb3

Crc4 – разрешить проверку и генерацию crc4 (не обязательный параметр)

Пример конфигурации для 4-х канального сервера:

span=1,1,0, ccs, hdb3, crc4

span=2,1,0, ccs, hdb3, crc4

span=3,1,0, ccs, hdb3, crc4

span=4,1,0, ccs, hdb3, crc4

14.2. Канальные интервалы для передачи данных и сигнализации

Каждому порту E1 соответствует 31 канальный интервал (TS0 отвечает за формат фрейма и в передаче данных не участвует). Нумерация TS в системе сквозная – для порта 1 соответствуют TS1..TS31, для порта 2 – TS32..TS62 и т.д. При конфигурации системы необходимо указать, какие каналы будут использоваться для передачи голосовых данных (B каналы), а какие будут использоваться для сигнализации (D канал). Каналы задаются ключевыми словами **bchan** и **dchan**.

Файл с конфигурацией	Параметр
/etc/dahdi/system.conf	bchan=<S>.-<E> dchan=<N>

где

S – номер начального TS группы,

E – номер конечного TS

В системе может быть несколько D каналов и множество B каналов. При их задании необходимо помнить, что используется сквозная нумерация.

Пример конфигурации:

bchan=1-15,17-31

dchan=16

14.3. Закон кодирования

Передача голосовых данных через порт E1 может осуществляться с использованием A или m закона кодирования. Выбор закона осуществляется ключевыми словами **alaw** или **mlaw**.

Файл с конфигурацией	Параметр
/etc/dahdi/system.conf	alaw=<S>.-<E> mlaw=<S>.-<E>

где

S – номер начального TS группы,

E – номер конечного TS

14.4. Режим локализации

Частоты и стиль сигналов «занято», КПВ и других сигналов, подаваемых сервером абоненту, определяется локализацией. Например, для России рекомендуется использовать следующие параметры:

Файл с конфигурацией	Параметр
/etc/dahdi/system.conf	loadzone=fr defaultzone=fr

14.5. Коммутация каналов

Группы канальных интервалов могут коммутироваться между собой. Эта опция задается ключевым словом `dacs`:

Файл с конфигурацией	Параметр
/etc/dahdi/system.conf	<code>dacs=<S>-<E>:<D></code>

где

S – номер начального TS группы,

E – номер конечного TS группы,

D – TS назначения

Коммутация каналов и соответствующей им CAS сигнализации задается ключевым словом `dacsrbs`.

Примеры:

dacs=3-5:10

означает, что канальные интервалы с 3,4,5 коммутруются с 10,11,12 без коммутации CAS сигнализации.

dacsrbs=3-5:10

означает, что канальные интервалы с 3,4,5 коммутруются с 10,11,12 с коммутацией CAS сигнализации.

14.6. Утилита *ecfg*

Утилита *ecfg* (*emcfg* для 1 и 2-х канальной версии сервера) может быть использована как простой анализатор E1. Утилита *ecfg* запускается со следующими параметрами в командной строке Linux:

```
# ecfg -i N
```

где

N – номер порта E1, начиная с 0 [0..7]

После запуска *ecfg* на экране отображается главное меню, где содержится информация о версии прошивки платы E1, а также статус выбранного порта E1.

```
Quasar monitor v.1.14 26/08/2008 Updates: http://parabel.ru/  
PMC/chan=0/0, conf. file="/etc/emcfg/quasarm0_0.cfg"  
HW/FW/REV version=10/10/e, driver verision=2.0.3
```

```
Line status: LOS=On , AIS=Off  
Frame status: LOF=On , Sa4..8=00000, RAIS=Off  
CAS Multiframe: CAS LOM=Off, XYXX=0000  
CRC4 Multiframe: CRC4 err=Off, LOC=On , E bit=On  
Err counters: HDB3=0, FAS=0, CRC4=0  
ABCD status: 00000000 00000000 00000000 00000000
```

```
1. Configuration >>  
2. Status >>  
3. Test >>  
0. Quit
```

14.7. Сигнализация ISDN PRI

14.7.1. Конфигурация

Перед настройкой сигнализации ISDN PRI, необходимо сконфигурировать порт E1 для работы с типом сигнализации CCS (см. стр. 29). Также, должны быть заданы параметры dchan и vchan (стр. 30).

ISDN PRI имеет несколько реализаций. Какая именно реализация будет использоваться, указывается в параметре **switchtype**.

Файл с конфигурацией	Параметр
/etc/asterisk/chan_dahdi.conf	switchtype=<type>

где type может принимать значения:

switchtype	Реализация протокола
national	National ISDN 2
dms100	Nortel DMS100
4ess	AT&T 4ESS
5ess	Lucent 5ESS
qsig	Q.SIG
euroisdn	EuroISDN (ETSI ET/5 или primary-net5)

Необходимо выбирать реализацию протокола того производителя, с которым предполагается работать. Если производитель неизвестен, выбирается протокол euroisdn.

В ISDN PRI также есть понятие типа оконечного оборудования – CPE или NET. Обычно на стороне оператора оборудование работает в режиме NET, на стороне абонента – в режиме CPE. Сервер может работать в обоих режимах, в зависимости от параметра **signalling**.

Файл с конфигурацией	Параметр
/etc/asterisk/chan_dahdi.conf	signaling=pri_cpe signaling=pri_net

14.7.2. Диагностика протокола

Диагностировать работу протокола можно в командной строке Asterisk. Для начала нужно подсоединиться к процессу Asterisk (из консоли сервера):

```
# asterisk -rvvvv
```

Следующие команды полезны для отладки соединений:

Команда	Выдаваемая информация
*CLI> pri show spans	Выводит список портов E1 (span), работающих в режиме PRI
*CLI> pri show span N	Выводит состояние сигнализации на порте (span) N – номер D канала, тип сигнализации, состояние счетчиков ошибок, значения внутренних таймеров
*CLI> pri debug span N	Включает распечатку ошибочных ситуаций на порте N.
*CLI> pri intense debug span N	Включает подробную распечатку всех событий на порте N.

14.8. Сигнализация SS7

Раздел в разработке.

15. Конфигурация FXO/FXS каналов

Настоящий раздел относится только к изделию Asteroid-MV16.

Сервер может обслужить до 128 FXO или FXS каналов, что соответствует 16 модулям FXS или FXO. Каждый модуль, устанавливаемый в систему, обслуживает 8 каналов одного типа – FXO или FXS. Тип телефонного окончания должен быть описан при установке модуля в файле `/etc/dahdi/system.conf`. Модули FXS должны быть описаны ключевым словом `fxsls`, модули FXO – ключевым словом `fxols`. За ключевыми словами `fxols` или `fxsls` должно следовать перечисление каналов данного типа. Соответствие номеров каналов слотам фиксировано и приведено в таблице.

Слот	Номера каналов (в системе без портов E1)	Номера каналов (в системе с E1 портами)
0	1-8	63-70
1	9-16	71-78
2	17-24	79-86
3	25-32	87-94
4	33-40	95-102
5	41-48	103-110
6	49-56	111-118
7	57-64	119-126
8	65-72	127-134
9	73-80	135-142
10	81-88	143-150
11	89-96	151-158
12	97-104	159-166
13	105-112	167-174
14	113-120	175-182
15	121-128	183-190

Если в системе установлены E1 порты, в описании появляется дополнительно 62 канала (2 порта E1, 31 канал каждый). В результате, как видно из таблицы, начальный канал соответствующего слота смещается на 62. Предположим, в слоте 9 установлен один модуль FXO, остальные позиции заняты модулями FXS. Тогда конфигурация должна задаваться параметрами:

Файл с конфигурацией	Параметр
/etc/dahdi/system.conf	fxsls=73-80 fxols=1-72,81-128

Прочие настройки аналоговых портов (закон кодирования, зона, адрес TDMoE контроллера) являются предустановленными, их не рекомендуется менять.

16. Возможности IP АТС

16.1. Введение

Функционал IP АТС обеспечивается программным обеспечением под управлением Asterisk – открытого программного проекта, который широко известен во всем мире. Привлекательность Asterisk обусловлена гибкостью конфигурации; наличием большого числа расширений и приложений; наличием технических специалистов, знакомых с данным программным продуктом. Ниже приведен далеко не полный список возможностей Asterisk.

- Настраиваемый план соединений
- Запись разговоров
- Голосовая почта
- Конференции (в том числе «виртуальные комнаты»)
- Парковка вызова
- Передача вызова по эстафете
- Очереди соединений
- Группы звонков

16.2. Конфигурационные файлы Asterisk

Процесс Asterisk стартует автоматически при запуске сервера. Доступ к командной строке Asterisk осуществляется из командной строки Linux:

```
# asterisk -r
```

В командной строке пользователь может наблюдать текущую конфигурацию IP АТС, а также выполнять различные команды диагностики.

Конфигурация IP АТС сохраняется в директории `/etc/asterisk` в виде текстовых файлов с определенным синтаксисом. Изменение конфигурации осуществляется пользователем с помощью любого текстового редактора, например входящего в состав файлового менеджера `mc`. После изменения конфигурации процесс Asterisk необходимо перезапустить, что можно сделать с помощью последовательности команд:

```
# /etc/init.d/asterisk stop
```

```
# /etc/init.d/asterisk start
```

Перечислим наиболее важные конфигурационные файлы.

asterisk.conf

Описывает расположение компонентов Asterisk, например, звуковых файлов, используемых для мелодии удержания вызова, скриптов (AGI), и других служебных файлов, используемых Asterisk.

extensions.conf

Содержит всеобъемлющий план соединений, обстоятельное описание пользователей IP АТС, голосовых почтовых ящиков, инструкций по обработке вызова, и описание поведения по умолчанию.

features.conf

Описывает дополнительные опции, например парковку вызова.

h323.conf

Описывает конфигурацию VOIP протокола H.323.

iax.conf

Описывает конфигурацию протокола IAX – для соединения нескольких Asterisk.

manager.conf

Описывает ограничения прав доступа для Asterisk's Manager Socket API.

mgcp.conf

Описывает поддержку в Asterisk протокола MGCP.

modules.conf

Содержит перечисление модулей (приложений), загружаемых при старте Asterisk.

sip.conf

Описывает конфигурацию VOIP протокола SIP.

logger.conf

Описывает расположение log файлов и уровень их подробности.

voicemail.conf

Содержит конфигурацию встроенного в Asterisk сервера голосовой почты.

Поведение Asterisk определяется также набором других конфигурационных файлов. Для управления сервером, как минимум, необходимо ознакомиться с файлами *extensions.conf*, *sip.conf*, *h323.conf*.

В большинстве, файлы в директории */etc/asterisk* следуют одинаковому синтаксису. Многие файлы имеют заголовки секций, которые начинаются со строки текста (заголовка) в квадратных скобках. Каждая секция содержит список конфигурационных установок. Установкой является строка (наименование параметра) за которой следует знак равенства (=) и устанавливаемое пользователем значение, например:

```
writeprotect=yes
```

16.3. Источники информации по IP ATC Asterisk

По конфигурации Asterisk написан ряд подробнейших документов, дублирование которых в данном руководстве было бы нецелесообразным. Документы помещены на CD диске, см. [8,9,10] в списке ссылок.

Приложение 1. Дополнительная литература

На CD с документацией прилагаются следующие статьи в русском переводе:

1. [Энциклопедия сетевого администратора Linux, версия 2.0](#)
2. [The Linux Networking Overview HOWTO](#)
3. [Сетевая поддержка в Линуксе, Linux NET-3-HOWTO.](#)
4. [Iptables Tutorial 1.1.19](#)
5. [Linux Advanced Routing & Traffic Control HOWTO](#)
6. [Quagga - расширенный пакет программ маршрутизации](#)
7. [Linux Bridge](#)
8. [Switching to VoIP. By Theodore Wallingford.](#)
9. [Asterisk Configuration Guide](#)
10. [The Asterisk Handbook](#)