Linux.yaroslavl.ru

Справочное руководство по Fetchmail.

ИМЯ

fetchmail - прием почты с серверов POP, IMAP, ETRN, или ODMR

СОДЕРЖАНИЕ

КРАТКИЙ ОБЗОР

fetchmail [*опция...*] [*сервер...*]

fetchmailconf

ОПИСАНИЕ

fetchmail - это утилита приема и пересылки почты; она забирает почту с удаленных почтовых серверов и передает ее локальной системе доставки почты на машине клиента. После этого вы можете читать почту обычными почтовыми агентами, например, *mutt*, *elm* или *Mail*. Утилита *fetchmail* может быть запущена в режиме демона для регулярной проверки и приема почты с одного или нескольких серверов.

fetchmail может забирать почту с серверов, поддерживающих любой протокол приема почты: POP2, POP3, IMAP2bis, IMAP4, IMAPrev1. Он также поддерживает расширения ESMTP ETRN и ODMR.

Хотя *fetchmail* проектировался для использования на периодических сеансах TCP/IP (например, соединения по протоколам SLIP или PPP), он может использоваться для доставки почты в защищенных системах, где запрещены соединения по SMTP с sendmail.

Каждое сообщение, принятое *fetchmail*, затем обычно пересылается по SMTP на порт 25 локальной машины, на которой он запущен (localhost), как будто это сообщение было принято по обычному каналу TCP/IP. Затем почта доставляется локально системным транспортным агентом (MDA - Mail Delivery Agent, обычно это sendmail, но можно использовать *smail*, *mmdf*, *exim*, *qmail*). Все механизмы управления доставкой (например, файлы *.forward*), обрабатываются системным MDA, и после этого срабатвают агенты локальной доставки почты.

Если порт 25 не доступен, а fetchmail знает о надежном локальном MDA, то этот MDA и будет использован для локальной доставки. Вот время сборки fetchmail обычно ищет исполняемые файлы. **procmail** и **sendmail**

Если у вас есть программа *fetchmailconf*, она может помочь в настройке и редактировании конфигурационного файла fetchmailrc. Она работает под X и требует установленных Python и Tk. Если вы настраиваете fetchmail для одного пользователя, выберите режим Novice. Режим Expert предоставляет полный контроль в настройке fetchmail, включая управление доменными (maildrop) ящиками. В любом случае, кнопка Autoprobe определит наилучший протокол, который поддерживает ваш почтовый сервер и предупредит о потенциальных проблемах.

ОПЕРАЦИИ

Поведение *fetchmail* управляется опциями командной строки и кофигурационным файлом, *~/.fetchmailrc*, синтаксис которого будет рассмотрен ниже (именно этот файл редактирует программа *fetchmailconf*). Опции командной строки имеют более высокий приоритет над опциями в *~/.fetchmailrc*.

При запуске fetchmail запрашивается каждый сервер, указанный в командной строке. Если сервера в командной строке не указаны, их имена берутся из файла *~/.fetchmailrc*.

Для облегчения использования *fetchmail* в скриптах и потоках, он возвращает код завершения.

Большинство опций имеют соответствующее ключевое слово, используемое в файле *fetchmailrc*.

Здесь не охвачены некоторые специфические опции; они перенесены в разделы "Аутентификация" и "Режим демона".

Общие опции

-V, --version

Вывод информации о вашей текущей копии *fetchmail.* Прием почты не производится. Вместо этого для каждого из указанных серверов выводятся все опции. Непечатные символы в паролях и прочих строках выводятся в C-стиле как ESC-последовательности,с обратной чертой. Эта опция полезна для проверки правильности ваших настроек.

-c, --check

Возвращает код состояния, оповещающего о наличии почты. Прием почты не производится (см. Коды возврата). Эта опция выключает режим демона. Она не очень хорошо работает с запросами нескольких серверов и не работает с ETRN или ODMR. Она возвращает ложный положительный ответ, если осталась прочтенная, но не удаленная почта в почтовом ящике на сервере, а ваш почтовый протокол не различает новые и прочитанные сообщения. Это значит, что эта опция будет работать с протоколом IMAP, не будет с POP2 и может сбоить на протоколе POP3.

-s, --silent

Тихий режим. Подавляет весь вывод сообщений о процессе/состоянии, которые обычно выводятся на stderr во время приема почты (это не влияет на сообщения об ошибках). Опция --verbose переопределяет этот режим.

-v, --verbose

Информативный режим. Все управляющие сообщения, проходящие между *fetchmail* и почтовым сервером, дублируются на stdout. Двойная опция (-v -v) увеличивает объем выдаваемой диагностики.

Опции приема почты

-a, --all (Ключ: fetchall) Прием как старых (прочитанных), так и новых сообщений с сервера. По умолчанию принимаются только непрочитанные сообщения. В POP3 эта опция, помимо прочего, выдает команду RETR вместо TOP. При использовании POP2 прием осуществляется так, как будто опция --all всегда включена. Эта опция не работает с ETRN или ODMR.

-k, --keep

(Ключ: keep) Сохраняет принятые сообщения на удаленном почтовом сервере. Обычно после приема сообщения удаляются с сервер. Опция **keep** сохраняет принятые сообщения в вашем почтовом ящике на почтовом сервере. Эта опция не работает с ETRN или ODMR.

-K, --nokeep

(Ключ: nokeep) Удаляет принятые сообщения с удаленного почтового сервера. Может быть полезна, если вы указали по умолчанию опцию **keep** в файле *.fetchmailrc*. Эта опция принудительно включается при использовании ETRN или ODMR.

-F, --flush

Только для POP3/IMAP. Удаляет старые (ранее прочитанные) сообщения с почтового сервера перед началом приема новых сообщений. Эта опция не работает с ETRN или ODMR. Внимание: Если ваш MTA завис и fetchmail аварийно завершил работу, при следующем запуске fetchmail удалит почту, которая так и не была вам доставлена. Лучше оставить настройки по умолчанию: если вы не указали опцию ‘-k’, то fetchmail автоматически удалит сообщения после их успешного приема.

Опции протоколов и запроса

-p, --protocol <proto>

(Ключ: proto[col]) Указывает протокол, который будет использоваться для связи с удаленным почтовым сервером. Если протокол не указан, принимается значение по умолчанию - AUTO. *proto* может принимать одно из следующих значений:

AUTO	Пробует IMAP, POP3, и POP2 (пропуская те, поддержка которых которых не была прикомпилирована).
POP2	Post Office Protocol 2
POP3	Post Office Protocol 3
APOP	POP3 с аутетификацией MD5 в старом стиле.
RPOP	POP3 с аутентификацией RPOP.
KPOP	POP3 с аутентификацией Kerberos V4 через порт 1109.
SDPS	POP3 с Demon Internet’s SDPS extensions.

IMAP IMAP2bis, IMAP4, или IMAP4rev1 (*fetchmail* сам определяет).

ETRN Использовать опцию ESMTP ETRN.

ODMR Использовать профиль On-Demand Mail Relay ESMTP.

Все эти протоколы работают по одному принципу - связываются с сервером и забирают почту, лежащую в почтовом ящике на сервере (кроме ETRN и ODMR). В режиме ETRN вы запрашиваете соответствующий ESMTP-сервер (например, BSD sendmail версий 8.8.0 и выше) для открытия SMTP-соединения с клиенской машиной и направления почты, лежащей в почтовой очереди сервера, на вашу машину. Режим ODMR работает аналогично ETRN, но не требует наличия на клиенской машине статического DNS.

-U, --uidl

(Ключ: uidl) Заставляет использовать UIDL (только в POP3), т.е. следить за "новизной" сообщений (UIDL означает ‘‘unique ID listing’’ и описан в RFC1725). Используется совместно с ‘keep’ при использовании почтового ящика несколькими пользователями.

-P, --port <portnumber>

(Ключ: port) Опция port определяет порт TCP/IP-соединения. Эта опция редко используется, т.к. все протоколы имеют свои стандартные номера портов.

--principal <principal>

(Ключ: principal) Опция principal позволяет указать ведущий сервис для взаимной аутентификации. Используется в POP3 или IMAP с аутентификацией Kerberos.

-t, --timeout <seconds>

(Ключ: timeout) Устанавливает предельное время "молчания" сервера в секундах. Если сервер не посылает приветственного собщения или не отвечает на команды в течении указанного времени, *fetchmail* "зависает". Если не указывать тайм-аут, то *fetchmail* может зависнуть навсегда, пытаясь получить почту с зависшего сервера. Это может быть весьма нежелательно, если *fetchmail* работает в фоновом режиме. Значение тайм-аута по умолчанию можно узнать, запустив fetchmail -V. Если соединение получает слишком много тайм-аутов, fetchmail будет считать такое соединение сбойным и прекратит прием почты, а пользователь получить об этом соответствующее письмо.

--plugin <command>

(Ключ: plugin) Опция plugin позволяет использовать внешнюю программу для установления TCP-соединения. Она полезна, например, при использовании SOCKS, SSL, ssh или в случае специальный настроек брандмауэра. Программа ищется в $PATH; в качестве параметров ей могут быть переданы аргументы "%h" и "%p". Fetchmail передает данные в программу через stdin, а получает из программы через stdout.

--plugout <command>

(Ключ: plugout) Аналог опции plugin, но используется для соединений SMTP.

-r <name>, --folder <name>

(Ключ: folder[s]) Указывает специфический почтовый ящик на почтовом сервере (или список ящиков через запятую), из которого надо взять почту. Синтаксис имени ящика зависит от сервера. Эта опция не доступна в POP3, ETRN или ODMR.

--tracepolls

(Ключ: tracepolls) Заставляет выводить информацию в виде ‘polling %s account %s’ где %s заменяются именем удаленного пользователяи и меткой сервера. Это может использоваться при фильтрации почты.

--ssl (Ключ: ssl) Указывает, что соединение с почтовым сервером должно быть зашифровано через SSL. С сервером образуется защищенное соединение. SSL должен быть запущен на сервере. Если не указан порт, то соединение устанавливается через порт SSL. Обычно этот номер порта отличается от номеров портов почтовых протоколов. Например, для IMAP используется 143, а для SSL - 993.

--sslcert <name>

(Ключ: sslcert) Определяет имя файла сертификата SSL клиента. Некоторые серверы, использующие SSL, при аутентификации требуют у клиента наличия ключей и сертификатов. В большинстве случаев это опциональный параметр. Он определяет местонахождение файла сертификата открытого ключа, который следует представить серверу во время установления SSL-соединения. Некоторые серверы могут затребовать такой сертификат.

--sslkey <name>

(Ключ: sslkey) Определяет имя файла личного ключа SSL клиента. Некоторые серверы, использующие SSL, при аутентификации требуют у клиента наличия ключей и сертификатов. Этот параметр определяет файл ключа, используемого для подписи транзакций с сервером во время установления соединения SSL. Он не нужен, если сервер не требует его, но некоторые серверы могут потребовать его наличия. Если ключ защищен паролем, то он будет запрошен при установлении соединения с сервером. Это может вызвать проблемы в режиме демона.

--sslproto <name>

(Ключ: sslproto) Указываетна использование протокола ssl. Возможные значения: '**ssl2**', '**ssl3**' и '**tls1**'. Попробуйте эту опцию, если стандартная процедура регистрации не проходит на вашем сервере.

--sslcertck

(Ключ: sslcertck) Указывает fetchmail строго сверить сертификат сервера со своими локальными доверенными сертификатами (см. опцию **sslcertpath**). Если сертификат сервера не подписан одним из доверенных (прямо или косвенно), соединение SSL не будет установлено. Эта проверка предупреждает атаки типа "промежуточного лица" в соединениях SSL. Обратите внимание, что CRL, видимо, в настоящее время не поддерживаются в OpenSSL при проверке сертификатов! При использовании этой опции ваши системные часы должны быть очень аккуратно настроены!

--sslcertpath <directory>

(Ключ: sslcertpath) Указывает каталог, в котором fetchmail ищет локальные сертификаты. По умолчанию - это каталог, принятый по умолчанию для OpenSSL. Помните, что всякий раз, когда вы добавляете или меняете сертификаты, вы должны запускать утилиту **c_rehash** для создания хеша.

--sslfingerprint

(Ключ: sslfingerprint) Определяет отпечаток ключа сервера (хеш MD5 ключа) в 16-ричной нотации, причем группы по 2 цифры разделяются двоеточиями. Буквы в 16-ричных числах должны быть в верхнем регистре. Этот формат принят OpenSSL, fetchmail использует его при установлении соединения SSL. Если ключи не совпадают, соединение не устанавливается.

Опции доставки

-S <hosts>, --smtphost <hosts>

(Ключ: smtp[host]) Задает список узлов, на которые следует пересылть почту (один или несколько имен узлов, разделенные запятыми). Узлы будут просматриваться в порядке их следования в списке. Первый откликнувшийся и будет использован для пересылки. Обычно в конец списка невидимо добавляется "localhost". Однако, при использовании Kerberos, в конец списка добавляется FQDN машины, на котрой запущен fetchmail. После каждого имени через косую черту можно указать номер порта. По умолчанию 25 (или ''smtp'' в IPv6). При указании абсолютного маршрута (т.е. начинающегося с /), он интерпретируется как имя UNIX-сокета, принимающего соединения LMTP (например, Cyrus IMAP daemon). Например:

```
--smtphost server1,server2/2525,server3,/var/imap/socket/lmtp
```

Эта опция может использоваться с ODMR, тогда fetchmail будет выполнять роль шлюза между серверами ODMR и SMTP.

--fetchdomains <hosts>

(Ключ: fetchdomains) В режиме ETRN или ODMR эта опция определяет список доменов, для которых будет приниматься почта, когда соединение сменит

направление. По умолчанию - это полное доменное имя (FQDN) машины, на которой запущен fetchmail.

-D <domain>, --smtpaddress <domain>

(Ключ: smtpaddress) Определяет домен, который добавляется к адресу в строках RCPT TO, посылаемых на SMTP-сервер. Если не указано, то используется имя SMTP-сервера (указываемого опцией --smtphost или "localhost" по умолчанию).

--smtpname <user@domain>

(Ключ: smtpname) Указывает имя домена и пользователя, которые помещаются в строку RCPT TO, передаваемой на SMTP-сервер. По умолчанию подставляется имя локального пользователя.

-Z <nnn>, --antispam <nnn[, nnn]...>

(Ключ: antispam) Определяет список кодов ошибок SMTP, которые интерпретируются как блокировка спама от сервера. Значение -1 отключает эту опцию. В командной строке список значений разделяется запятыми.

-m <command>, --mda <command>

(Ключ: mda) С помощью этой опции вы можете передать почту непосредственно доставочному агенту (MDA) минуя порт 25. Для избежания потери почты используйте эту опцию только с sendmail или procmail, которые возвращают коды завершения в случае возникновения ошибок; ненулевой код возврата вынуждает fetchmail прекратить прием почты и оставить сообщение на сервере. Если *fetchmail* выполняется под бюджетом root, он устанавливает свой userid в имя того пользователя, которому доставляет почту. Возможные MDA - `"/usr/sbin/sendmail -oem -f %F %T"`, `"/usr/bin/deliver"` и `"/usr/bin/procmail -d %T"` Адреса локальной доставки подставляются вместо %T, адрес отправителя из поля заголока From - вместо %F. *Не* используйте вызов MDA в виде "sendmail -oem -t", который обрабатывает содержимое To/Cc/Bcc, это может вызвать зацикливание почты и гнев многих администраторов почты.

--lmtp (Ключ: lmtp) Указывает на доставку через LMTP (Local Mail Transfer Protocol). В этом случае *необходимо* указать номер порта (через косую черту) для каждого узла в списке опции smtphost. Номер порта по умолчанию 25 указывать нельзя.

--bsmtp <filename>

(Ключ: bsmtp) Добавляет принятую почту в файл BSMTP. Он просто содержит команды SMTP, которые fetchmail обычно посылает демону SMTP в процессе пересылки почты. Аргумент ‘-’ вызывает вывод почты на стандартный вывод. Учтите, что реконструкция команд MAIL FROM и RCPT TO не может быть гарантированно правильно. Особенно это касается доменных ящиков. Эти предупреждения описаны в разделе "Использование доменных ящиков"

Опции управления ограничениями

-l <maxbytes>, --limit <maxbytes>

(Ключ: limit) Устанавливает максимальный размер письма в байтах. Сообщения размером больше указанного не будут приниматься, но будут оставлены на сервере (если fetchmail работает не в фоновом режиме, то такие сообщения будут отмечены как "oversized"). Если протокол примеа допускает (например, IMAP и POP3 без опции fetchall), сообщения не будут помечены как прочтенные. Установка значения в 0 снимает ограничения на размер, установленные в файле конфигурации. Эта опция полезна для ограничения приема больших сообщений в дневные часы с высоким телефонным тарифом. При работе в режиме демона пользователю посылается соответствующее письмо. Эта опция не работает с ETRN и ODMR.

-w <interval>, --warnings <interval>

(Ключ: warnings) Значение параметра - интервал в секундах. При вызове fetchmail с опцией --limit в режиме демона, эта опция определяет интервал, с которым пользователю посылаются письма с уведомлением о наличии большого письма.

-b <count>, --batchlimit <count>

(Ключ: batchlimit) Определяет максимальное число сообщений, которое будет отправлено по SMTP до того, как соединение будет закрыто и открыто вновь (0 означает отсутствие ограничения). Явное указание --batchlimit в командной строке переопределяет ограничения в конфигурационном файле. Тогда как sendmail обычно начинает доставку сообщений сразу после получения конца сообщения, другие MTA, типа qmail и smail, могут ждать закрытия сокета. Это вызывает задержки при обработке очень больших пакетов. Указание этой опции позволяет устранить такие задержки. Эта опция не работает с ETRN и ODMR.

-B <number>, --fetchlimit <number>

(Ключ: fetchlimit) Ограничивает число сообщений, принимаемых с сервера за один сеанс. По умолчанию такого ограничения нет. Эта опция не работает с ETRN и ODMR.

-e <count>, --expunge <count>

(Ключ: expunge) Вызывает окончательное удаление (вычеркивание) после приема указанного числа сообщений. При использовании POP2 или POP3 невозможно сделать окончательное удаление, пока не будет дана команда QUIT в конце сессии. При использовании этой опции fetchmail разделит длинный сеанс приема почты на несколько сессий, посылая QUIT в конце каждой из них. Это очень удобно в случае частых разрывов связи при плохом качестве телефонной линии. При использовании IMAP, fetchmail посылает команду EXPUNGE после каждого удаления. Это очень полезно, если ваше соединение нестабильно - исключается появление дубликатов сообщений. Однако, на больших почтовых ящиках это вызывает частую переиндексацию и перегржает сервер, поэтому применяйте эту опцию не слишком часто. Если вы указываете значение этой опции как целое N, то *fetchmail* будет вызывать expunge после каждого N-го удаления. Значение 0 отменяет expunge. Эта опция не работает с ETRN и ODMR.

Опции аутентификации

-u <name>, --username <name>

(Ключ: user[name]) Указывает имя пользователя, используемого для авторизации на удаленном почтовом сервере. По умолчанию - это имя пользователя локальной машины, на которой запущен fetchmail.

-I <specification>, --interface <specification>

(Ключ: interface) Проверяет, чтобы перед приемом почты был активен указанный интерфейс и имел указанный локальный или удаленный адрес (или диапазон). Часто *fetchmail* используется по телефонным каналам по протоколам SLIP или PPP. Это относительно безопасный канал. Но если к почтовому серверу существуют другие маршруты (например, если связь устанавливается через альтернативного провайдера), ваши имя и пароль могут быть перехвачены (особенно в режиме демона с периодическим приемом почты). Опция --interface помогает это предотвратить. Если указанное соединение не активно, или имеет некорректный адрес, прием почты не осуществляется. Формат опции следующий:

```
interface/iii.iii.iii.iii/mmm.mmm.mmm.mmm
```

Поле перед первой косой чертой (`interface`) - имя интерфейса (например, ppp0). Второе поле (`iii.iii.iii.iii`)- это допустимый IP-адрес. Третье поле (`mmm.mmm.mmm.mmm`) - это маска, задающая диапазон адресов. Если она не указана, подразумевается маска 255.255.255.255 (т.е. точное совпадение). Эта опция поддерживается только в Linux и FreeBSD. Специфичную информацию для FreeBSD см. в опци **monitor**

-M <interface>, --monitor <interface>

(Ключ: monitor) Режим демона может занять линию, которая должна отключаться по истечении периода неактивности (например, PPP), на бесконечное время. Эта опция заставляет следить за активностью интерфейса TCP/IP. Перед каждым приемом почты делается проверка линии - если интерфейс поднят и не наблюдается другой активности, то прием пропускается. Однако, если fetchmail "просыпается" по сигналу, то проверка интерфейса не производится, и прием присходит в любом случае. Эта опция поддерживается только в Linux и FreeBSD. Чтобы опции **monitor** и **interface** работали в FreeBSD

под непривилегированным пользователем, исполняемый файл fetchmail должен быть установлен SGID kmem. Возможно, это брешь в безопасности, но fetchmail работает под эффективным GID, установленным в эту kmem group *только* во время собирания данных с интерфейса.

--auth <type>

(Ключ: auth[enticate]) Определяет тип аутентификации. Возможные значения: **any**, '**password**', '**kerberos_v5**' и '**kerberos**' (точнее, '**kerberos_v4**'), gssapi, *cram-md5*, *otp*, *ntlm*, and **ssh**. Если указано значение **any** (по умолчанию), fetchmail пробует сначала методы, не требующие пароля (GSSAPI, KERBEROS_IV); затем смотрит методы, скрывающие ваш пароль (CRAM-MD5, X-OTP, NTLM), и если сервер не поддерживает вышеперечисленное, используются методы, передающие пароль в открытом виде. Другие значения могу использоваться для разных методов аутентификации (**ssh** подавляет аутентификацию). Любое значение, отличное от *password*, *cram-md5*, *ntlm* или *otp* не требует пароля. Укажите *ssh*, если вы используете защищенное соединение; gssapi или **kerberos_v4**, если используются протоколы, производные от GSSAPI или K4. Выбор протокола KPOP автоматически включает аутентификацию Kerberos. Эта опция не работает с ETRN.

Прочие опции

-f <pathname>, --fetchmailrc <pathname>

Определяет имя конфигурационного файла, отличного от *~/.fetchmailrc*. Вместо имени можно указать "-" (одиночное тире), что означает чтение настроек со стандартного ввода. Конфигурационный файл должен иметь права доступа не более 600 (u=rw,g=,o=), иначе fetchmail не запустится.

-i <pathname>, --idfile <pathname>

(Ключ: idfile) Определяет альтернативное имя файла `.fetchids` для хранения UID.

-n, --norewrite

(Ключ: no rewrite) Обычно *fetchmail* меняет заголовки (To, From, Cc, Bcc, и Reply-To) в принятой почте так, преобразуя их в полные адреса (@ и имя почтового сервера). Это облегчает делать ответы на письма (иначе все адресаты подразумевались бы как локальные пользователи вашей клиентской машины!). Эта опция запрещает переопределение адресов. (Эта опция включена по настоянию многих параноиков, осмелившихся редактировать правила подстановки sendmail; на самом деле это не очень хорошая идея отключать переопределение заголовков).

-E <line>, --envelope <line>

(Ключ: envelope) Эта опция изменяет заголовок, который fetchmail подразумевает как несущий копию адреса конверта, "X-Envelope-To". Но поскольку этот заголовок не стандартный, на практике всякое бывает. Подробнее смотрите ниже обсуждение доменных почтовых ящиков (multidrop). Как особый случай, 'envelope "Received"' разрешает обработку строк Received в стиле sendmail. Это установлено по умолчанию, и не следует менять, если вы глобально не отключили обработку строк "Received" опцией 'no envelope' в конфигурационном файле.

-Q <prefix>, --qvirtual <prefix>

(Ключ: qvirtual) Значением этой опции является строка-приставка, которая удаляется из имени пользователя, если оно найдено в заголовке, определенном в опции *envelope перед* преобразованием имен в случае доменного ящика или проверки локального домена. Эта опция полезна, если вы забираете почту для целого домена, а ваш провайдер использует qmail. Одной из особенностей qmail является заголовок сообщения

```
'Delivered-To:'
```

Когда qmail доставляет почту в локальный почтовый ящик, он помещает в эту строку имя пользователя и имя узла. Основная причина этого - предотвращение зацикливания почты. При настройке qmail для обработки почты для отключенной машины, почтовый сервер провайдера обычно помещает имя этой машины в свой файл 'Virtualhosts' с тем, чтобы он добавлялся в начало всех адресов для этого сайта. Это вызывает отправку почту к пользователю 'username@userhost.userdom.dom.com', имеющей строку 'Delivered-To:' в виде:

```
Delivered-To: mbox-userstr-username@userhost.userdom.dom.com
```

Провайдер может создать любой префикс 'mbox-userstr-', но обычно используется имя пользовательского узла. С помощью опции 'envelope Delivered-To:' fetchmail может надежно определить получателя, но префикс 'mbox-userstr-' необходимо отрезать. Для этого и предназначена эта опция.

--configdump

Проверяет файл *~/.fetchmailrc*, интерпретирует опции командной строки и создает отчет о конфигурации на стандартный вывод. Отчет представляет собой набор правил присвоений на языке Python. Эта опция предназаначена для использования в утилите интерактивной настройки *fetchmailconf*, написанной на Python.

АУТЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЕЙ И ШИФРОВАНИЕ

Во всех режимах, кроме ETRN, необходима аутентификация клиента. Обычная аутентификация пользователя в *fetchmail* сходна с механизмом, используемом в **ftp**. Проверка правильности идентификатора пользователя и пароля зависит от применяемой на почтовом сервере системы безопасности.

Если почтовый сервер является Unix-машиной, на которой вы имеете свою учетную запись, *fetchmail* будет использовать ваши обычные имя и пароль. Если вы используете одно и то же имя и на сервере, и на клиенте, вам можно не указывать идентификатор пользователя в опции **-u** -- по умолчанию используется ваше имя на клиентской машине. Если же на сервере используется другое, укажите его в опции **-u**. Например, если ваше учетное имя 'jsmith' на машине с именем 'mailgrunt', вы можете запсутить *fetchmail* следующим образом:

```
fetchmail -u jsmith mailgrunt
```

По умолчанию *fetchmail* интерактивно запрашивает ваш пароль перед установлением соединения. Это самый безопасный способ использования *fetchmail*; он обеспечивает, что ваш пароль не будет скомпрометирован. Однако, вы можете указать свой пароль в файле *~/.fetchmailrc*. Это полезно при запуске *fetchmail* в режиме демона или в скриптах.

Если вы не указали пароль, и *fetchmail* не обнаружил его в вашем файле *~/.fetchmailrc*, то будет предпринята попытка выудить пароль из файла *~/.netrc* в вашем домашнем каталоге, и только потом будет сделан интерактивный запрос. Первым делом Fetchmail смотрит на соответствие имени в опции poll; если не найдено, проверяется имя в via. Синтаксис файла *~/.netrc* описан в документации по **ftp**.

На почтовых серверах, не содержащих обычных бюджетов пользователей, ваш идентификатор и пароль обычно присваиваются администратором сервера при открытии почтового ящика. Если вы не знаете свой идентификатор и пароль для доступа к почтовому ящику, обратитесь к своему администратору.

Ранние версии POP3 (RFC1081, RFC1225) использовали грубую форму аутентификции с использованием на стороне сервера файла *rhosts*. В варианте RPOP на сервер через специальный порт подается эквивалент пароля, а серверу вместо команды PASS посылается команда RPOP, чтобы тот предпринял соответствующую проверку. RPOP поддерживается в *fetchmail* (укажите опцию 'protocol RPOP'), но настоятельно не рекомендуется использовать, поскольку с помощью спуфинга такой пароль легко перехватить.

RFC1460 проедлагает утентификацию APOP. В этом варианте POP3 вы регистрируете пароль APOP на сервере (для этого есть специальная программа *popauth*), и помещаете свой пароль в файл *~/.fetchmailrc*. При каждой регистрации *fetchmail* на сервере, он посылает зашифрованный хеш вашего пароля и времени сервера, который затем проверяется в базе данных.

Если *fetchmail* собран с поддержкой Kerberos, и вы указываете аутентификацию Kerberos (опцией --auth или опцией **authenticate kerberos_v4** в файле *.fetchmailrc*), fetchmail при каждом приеме почты пытается получить у почтового сервера билет Kerberos. Если в параметрах poll или via указано значение 'hesiod', то fetchmail

попытается использовать Hesiod для поиска сервера.

При использовании POP3 или IMAP с аутентификацией GSSAPI, *fetchmail* проверит, поддерживает ли почтовый сервер функции GSSAPI (описанные в RFC1731 и RFC1734), и в случае положительного результат будет их использовать. В настоящее время это проверено только для Kerberos V, поэтому одидается, что у вс уже есть нужный билет Kerberos. Вы можете передать имя пользователя, отличающееся от стандартного, через опцию **--user** или через ключевое слово **user** в *.fetchmailrc*.

Если демон IMAP во время аутентификаиции возвращает ответ PREAUTH, fetchmail распознает это и пропускает стандартный этап аутентификации. Это может быть полезно при запсуске imapd вместе с ssh. В этом случае вы можете объявить значение аутенификации ‘ssh’, это предупредит запрос у вас пароля при запуске fetchmail.

Если вы используете POP3, а сервер посылает одноразовый пароль в соответствии с RFC1938, *fetchmail* будет использовать ваш пароль как фразу для генерации ответа. Это предотвращает пересылку паролей в открытом виде.

Поддерживается также аутентификация Compuserve’s RPA (сходная с APOP). Если вы включили в *fetchmail* ее поддержку, то вместо посылки незашифрованного пароля производится аутентификация RPA, если в имени узла обнаруживается "@compuserve.com".

Если вы используете IMAP, то поддерживается также Microsoft’s NTLM (применяется в Microsoft Exchange). Если вы включили в *fetchmail* ее поддержку, fetchmail пробует провести аутентификацию NTLM (вместо посылки открытого пароля) всякий раз, когда сервер на запрос его возможностей отвечает AUTH=NTLM. Опция user указываетс в виде ‘user@domain’: левая часть до @ (*user*) - это имя пользователя, а правая часть (*domain*)- имя домена NTLM.

При использовании IPSec, можно указать опцию -T (--netsec) для передачи защищенного запроса, используемого при инициализации исходящего соединения IP. Значение этой опции - строка, передаваемая параметром в функцию net_security_strtorequest() библиотеки inet6_apps.

Доступ к функциям шифрации осуществляется указанием опции --ssl или ключевого слова "ssl" в файле .fetchmailrc. При включенном SSL все запросы посылаются после установления SSL-соединения. Некоторые сервисы,например, POP3 и IMAP, используют номера портов отличные, чем SSL. Номера защищенных портов выбираются автоматически.

При установлении соединения с сервером SSL, сервер предъявляет клиенту для проверки сертификат. Сертификат проверяется на предмет соответствия имени сервера в сертификате настоящему имени сервера, а также проверяетя срок действия сетрификата. Если любое из этих правил не подтверждается, выводится предупреждение, но соединение продолжается. Сертификат сервера не обязательно должен быть подписан специальной Службой Сертификации, это может быть "самоподписанный" сертификат.

Некоторые SSL-серверы могут затребовать сертификат клиента. В этом случае можно указать размещение сертификата и личного ключа. Сертификат клиента направляется на сервер для проверки. Если сертификат отсутствует или неверен, соединение может быть прекращено. Некоторые серверы также требуют, чтобы сертификаты были подписаны известной Службой Сертификации.

И, наконец, последнее слово об использовании SSL: хотя описанный метод с использованием самопальных сертификатов и защищает от пассивных наблюдателей, он не спасает от активных атак. Конечно, этот гораздо лучше пересылки открытых паролей, но вы должны всегда учитывать атаку "промежуточного звена" (например, такими утилитами, как dsniff http://www.monkey.org/~dugsong/dsniff/). Если вы беспокоитесь о безопасности вашего почтового ящика, используйте туннелирование ssh (ниже есть примеры).

РЕЖИМ ДЕМОНА

Опции **--daemon <interval>** или **-d <interval>** запускают *fetchmail* в режиме демона. В качестве interval укажите числовой аргумент - это интервал приема почты в секундах.

В режиме демона *fetchmail* переводит себя в фоновый режим и работает бесконечно,

запрашивая указанный узел, а затем засыпая на указанный период времени.

Напрмер, при зарпуксе

```
fetchmail -d 900
```

fetchmail будет опрашивать все почтовые сервера, пересисленные в вашем файле *~/.fetchmailrc* (кроме содержащих ключевое слово skip), каждые 15 минут.

Можно также указать интервал выборки в файле *~/.fetchmailrc*, задав опцию ‘set daemon <interval>’, где <interval> - это целое число секунд. В этом случае fetchmail всегда будет стартовать в режиме демона, если вы не переопределите это в командной строке опцией --daemon 0 или -d0.

В режиме демона возможен только один процесс на каждого пользователя, для обеспечения этого fetchmail создает файлы локировки.

Обычно вызов fetchmail с опцией daemon в фоновом режиме посылает сигнал демону, вызывая немедленный запрос почтового сервера. Для этого служит сигнал SIGHUP, если fetchmail запущен от root, или SIGUSR1 в противном случае. При этом также очищаются все флаги "заклинивания", установленные ранее при неудачной аутентификации или многличсленных таймаутах.

Опция **--quit** прекращает работу демона. Она используется только в командной строке.

Эта опция может быть использована совместно с другими опциями; она останавливает работу работающего демона до того, как сработают остальные опции командной строки в комбинации с настройками конфигурационного файла.

Опция **-L <filename>** или **--logfile <filename>** (ключ: set logfile) позволяет направить информационные сообщения в указанный лог-файл, имя которого указывается в качестве аргумента filename. Лог-файл открывается на добавление, поэтому предыдущие сообщения не удаляются. Эта опция используется в основном для отладки.

Опция **--syslog** (ключ: set syslog) задает направление всех информационных сообщений и сообщений об ошибках в демон **syslog**. Сообщения записываются с идентификатором **fetchmail**, уровнем **LOG_MAIL**, и приоритетом **LOG_ERR**, **LOG_ALERT** или **LOG_INFO**. Эта опция предназначена для записи состояния и сообщений об ошибках во время приема почты с сервера. Сообщения об ошибках командной строки или во время проверки *.fetchmailrc* все равно выдаются на stderr или в указанный файл журнала. Опция **--nosyslog** отключает использование syslog, подраземеная, что это было включено в *~/.fetchmailrc* или в командной строке через **-L** или **--logfile <file>**

Опция **-N** или **--nodetach** предотвращает перевод процесса в фоновый режим и открепление от управляющего терминала. Обычно используется при отладке. Учтите, что в этом случае опция logfile игнорируется.

При приеме почты в режиме демона с сервера POP2 или IPAP2bis, временные ошибки (например, сбой DNS или отказ sendmail доставить почту) могут принудительно включить опцию fetchall при следующем сеансе приема почты. Это означает, что если сообщение принято (и помечено на сервере как прочтенное), но локально не доставлено, в следующем сеансе приема почты оно будет приянто повторно (IMAP не удаляет сообщения с сервера, пока они не будут доставлены, так что там такой проблемы не возникает).

Если вы изменили или тронули файл *~/.fetchmailrc* во время работы демона, это обнаруживается при следующем сеансе приема почты; тогда fetchmail заново считывает настройки и перезапускает себя (используя exec())

АДИМИНИСТРАТИВНЫЕ ОПЦИИ

Опция **--postmaster <name>** (ключ: set postmaster) задает имя пользователя, которому будет направлена почта в случае, когда не удалось определить локального получателя. Обычно это пользователь, запустивший fetchmail. Если это root, то принимается значение ‘postmaster’. Пустая строка вызывает уничтожение подобной почты.

Опция **--nobounce** предотвращает обычную отправку сообщений об ошибках отправителю письма. Если nobounce включено (on), то сообщение посылается пользователю postmaster.

Опция **--invisible** (ключ: set invisible) пытается сделать fetchmail невидимым. Обычно

fetchmail выступает в роли обычного траспортного агента (MTA) - он моздает заголовок Received и указывает MTA, доставляющего почту, что она пришла от fetchmail. Если invisible включено (on), соответствующий заголовок Received не создается, и fetchmail пытается заставить агента доставки думать, что почта пришла непосредственно с почтового сервера.

Опция **--showdots** (ключ: set showdots) заставляет fetchmail рисовать точки (инидикатор прогресса), даже если текущий терминал не является stdout (например, лог-файл). Начиная с версии 5.3.0 точки выводятся только на stdout.

Указав опцию **--tracepolls**, вы можете заставить fetchmail добавить информацию в заголовок Received в виде "polling {label} account {user}", где {label} - это метка учетной записи из указанного конфигурационного файла, обычно ~/.fetchmailrc, {user} - имя пользователя, используемое для регистрации на почтовом сервере. Этот заголовк может быть полезен для сортировки почты по разным почтовым ящикам (например, в случае, когда вы получаете письмо с сервера, ведущего список рассылки, и подписаны на этот список рассылки). По умолчанию такого заголовка не добавляется. В файле *.fetchmailrc* ключевое слово ‘tracepolls’.

СБОИ ПРИ ПРИЕМЕ ПОЧТЫ

Протоколы, используемые *fetchmail* при общении с почтовыми серверами, очень надежны. Обычно при пересылке на порт 25 сообщения не удаляются (и даже не метятся на удаление) на сервере до тех пор, пока агент SMTP не даст fetchmail'у подтверждения о том, что сообщение принято к доставке или отвергнуто по причине спама.

Тем не менее, при пересылке на MDA всегда вероятны сбои. Некоторые MDA достаточно надежны и возвращают ненулевой код возврата в случае ошибки, даже по причине нехватки ресурсов. К таким принадлежат **procmail**, **sendmail**, **exim**. Эти программы возвращают надежное подтверждение и могут использоваться без риска потери почты. Небезопасные MDA могут возвращать 0 даже при сбое в доставке. В таком случае вы потеряете почту.

В обычном режиме fetchmail пытается принять только новые сообщения, оставить нетронутыми (и не удаляя) сообщения, которые вы прочитали непосредственно на сервере (или принятые в предыдущем сеансе fetchmail с опцией --keep). Однако, вы можете обнаружить, что прочтенные сообщения принимаютя и удаляются, даже если вы не указали опцию --all. Вот возможные причины этого.

Одной из них является использование протокола POP2. В этом протоколе не поддерживаются состояния сообщений ‘new’ или ‘old’, поэтому *fetchmail* считает все сообщения всякий раз новыми. Но это маловероятно, т.к. POP2 устарел и редко используется.

POP3 и RFC1725. В этой версии протокола POP3 нет команды LAST, но некоторые серверы POP3 следуют этому (это можно проверить, запустив "fetchmail -v"). *fetchmail* пытается компенсировать это использованием UID, сохраняя идентификаторы сообщений, просмотренных в каждом сеансе, в файле *.fetchids*. Но при этом не ведется список сообщений для остальных клиентов, а аткже сообщений, прочитанных мейлером непосредственно на сервере и не удаленных. Наилучшее решение этой проблемы - использовать протокол IMAP.

Еще одна потенциальная проблема в использовании POP3 заключается в том, что серверы могут вставлять сообщения в середину почтового ящика (ходят слухи, что некоторые реализации mail в VMS страдают этим недостатком). *fetchmail* подразумевает, что новые сообщения добавляются в конец почтового ящика; в противном случае некоторые старые сообщения могут считаться новыми и наоборот. Единственное средство решения этой проблемы - использовать IMAP.

Еще одна проблема с POP3 состоит в том, что если нельзя создать файл локировки в домашнем каталге пользователя, некотрые серверы POP3 вовращают недокументированный ответ, который заставляет fetchmail сообщать об отсутствии почты.

IMAP использует наличие или присутствие флага "Seen" для определения того, новое это сообщение или старое. В Unix серверы IMAP используют флаги состояния в стиле BSD, устанавливаемые пользовательскими агентами, и в случае необходимости устанавливают флаг "Seen". Все IMAP-серверы под Unix делают это, хотя это и не описано в RFC. Если вам попадется (крайне маловероятно) сервер, не делающий этого,

все старые сообщения будут казаться новыми. В этом случае сообщения, принятые fetchmail с опцией --keep, будут восстановлены и помечены как старые.

В режимах ETRN и ODMR, *fetchmail* на самом деле не принимает почту; вместо этого он запрашивает SMTP-сервер начать выдачу почтовой очереди клиента по SMTP. Таким образом, пересылаются только недоставленные сообщения.

ФИЛЬТРАЦИЯ СПАМА

Многие SMTP-серверы позволяют администраторам настраивать фильтрацию спама для блокировки нежелательной почты с определенных доменов. При этом происходит проверка строк MAIL FROM и DATA, и в случае обнаружения спама формируется SMTP-ответ (который, к сожалению, разный у разных серверов).

Последние версии *sendmail* возвращают код ошибки 571. Это значение зафиксировано в RFC 1893 как "Delivery not authorized, message refused" ("Доставка запрещена, сообщение отклонено").

В соответствии с текущими проектами замены RFC 812, правильным кодом должен быть 550 "Requested action not taken: mailbox unavailable" ("Запрошенное действие не выполнено: почтовый ящик не доступен") - (также добавляют "Почтовый ящик не найден, в доступе отказано или команда отвергнута политикой безопасности").

exim возвращает 501 "Syntax error in parameters or arguments", но также скоро перейдет на 550.

postfix возвращает 554.

fetchmail уничтожает сообщение, реагируя на любой код из списка [571, 550, 501, 554], но может быть настроен опцией ‘antispam’ Это один из трех случаев, когда fetchmail уничтожает почту (остальные два - это ошибки 552 и 553, и подавление multidropped сообщений с уже прочитанным message-ID).

При приеме почты с IMAP-сервера спам обнаруживается сразу по получении заголовка письма, и сообщение отвергается без приема тела письма. Поэтому вам не стоит беспокоиться об оплате времени, уходящего на приема всего сообщения.

Если включена опция *spambounce*, отправитель получит сообщение, информирующее его о том, что мы не принимаем от него почту.

ОБРАБОТКА ОШИБОК SMTP/ESMTP

Помимо обработки спама, fetchmail предпринимает определенные действия в случае получения следующих кодов ошибок

452 (insufficient system storage)

Оставляет сообщение в почтовом ящике на сервере для последующего приема.

552 (message exceeds fixed maximum message size)

Удаляет сообщение с сервера. Посылает уведомление отправителю.

553 (invalid sending domain)

Удаляет сообщение с сервера. Посылает уведомление отправителю.

При остальных ошибках письмо возвращается отправителю.

КОНФИГУРАЦИОННЫЙ ФАЙЛ

Наилучший способ настройки fetchmail состоит в написании файла *.fetchmailrc* в вашем домашнем каталоге (вы можете использовать обычный текстовый редактор или интерактивную утилиту *fetchmailconf*). Если возникает конфликт между опциями командной строки и опциями в файле, то предпочтение отдается командной строке.

Для защиты ваших паролей маска прав доступа к файлу *~/.fetchmailrc* не может быть больше 600 (u=rw,g=,o=); *fetchmail* проверяет это, и в противном случае прекращает работу.

Вы можете рассматривать *.fetchmailrc* как список команд, выполняемых *fetchmail*, если он вызывается без аргументов.

Синтаксис конфигурационного файла

Комментарии начинаются с ’#’ и продолжаются до конца строки. В остальном файл состоит из серий определений серверов или операторов установки глобальных опций в свободном формате на основе ключевых слов.

Существуют 4 типа ключевых слов: грамматические ключевые слова, числа, строки без кавычек и строки в кавычках. Строка в кавычках помещается в двойные кавычки и может содержать пробелы (цифры в кавычках считаются строкой). Строка без каычек - это ключевые слова, разделенные пробелами, они не должны быть числами, не должны содержать специальных символов ‘,’, ‘;’, ‘:’, или ‘=’.

Ключевые слова в определении серверов разделаются любым числом пробелов, в противном случае пробелы игнорируются. Можно использовать ESC-последовательности в стиле языка C (\n, \t,\b) для маскировки непечатных символов.

Строка определения сервера состоит из одного из ключевых слов ‘poll’ или ‘skip’, после которого идет имя сервера, затем опции сервера, затем информация о пользователе. Замечание: распространенная ошибка заключается в смешивании опций сервера и опций пользователя.

Для обратной совместимости, слово ‘server’ является синонимом ‘poll’.

Вы можете использовать слова ‘and’, ‘with’, ‘has’, ‘wants’, и ‘options’ в любом месте. Они служат лишь для облегчения читаемости файла и игнорируются. Символы пунктуации ’:’, ’;’ и ’,’ также игнорируются.

Poll и Skip

Слово ‘poll’ заставляет fetchmail запрашивать указанный сервер, если запускается без аргументов. ‘skip’ заставляет fetchmail не заправшивать указанный сервер, до тех пор пока он не будет указан в командной строке. (‘skip’ позволяет безопасно экпериментировать с разными определениями или запрещать сервера, которые временно не доступны).

Ключевые слова и опции

Здесь приведены доступные опции. Суффиксы, заключенные в квадратные скобки, не обязательны. Соответствие опции командной строки задается символом "-" и буковой опции.

Глобальные опции:

Ключевое слово	Опция	Функция
set daemon		Устанавливает прием почты в фоновом режиме через указанные интервалы времени
set postmaster		Имя пользователя, получающего почту, если не удалось найти подходящего получателя
set no bouncemail		Направалять ошибочную почту постмастеру, а не отправителю
set no spambounce		Отправлять спам обратно
set logfile		Имя лог-файла для регистрации ошибок и информационных сообщений
set idfile		Имя файла для хранения списка UID
set syslog		Направлять ошибки в syslog.
set nosyslog		Выключить направление ошибок в syslog.
set properties		Игнориуется fetchmail (может использоваться в скриптах)

Опции сервера:

Ключевое слово	Опция	Функция
via		Имя DNS почтового сервера, переопределяющее poll

proto[col]	-p	Протокол (регистр не важен): POP2, POP3, IMAP, APOP, KPOP
local[domains]		Домен(ы), считаемые локальными
port	-P	Порт TCP/IP
auth[enticate]		Тип аутентификации (по умолчанию ‘password’)
timeout	-t	Время тайм-аута в секундах (по умолчанию 300)
envelope	-E	Имя заголовка адреса конверта
no envelope		Запрет поиска адреса в конверте
qvirtual	-Q	Префикс имени домена Qmail, который надо удалить из имени пользователя
aka		Альтернативные имена DNS почтового сервера
interface	-I	IP-адрес интерфейса(-ов), который должен быть поднят перед началом приема почты
monitor	-M	Указывает IP-адрес для мониторинга активности
plugin		Определяет команду, через которую устанавливается соединение с сервером.
plugout		Определяет команду, через устанавливается соединение с SMTP.
dns		Разрешает поиск в DNS для доменного (multidrop) ящика (по умолчанию).
no dns		Отключить DNS для multidrop
checkalias		Сравнивать адреса IP для multidrop
no checkalias		Не сравнивать имена для multidrop (по умолчанию)
uidl	-U	Использовать клиентские UIDL для POP3
no uidl		Отключить использование UIDL в POP3 (по умолчанию)
interval		Проверять сервер каждые N секунд.
netsec		Передать опцию в запрос IPsec.
principal		Установить Kerberos principal (только при использовании imap и kerberos)

Опции пользователя:

Ключевое слово	Опция	Функция
user[name]	-u	Имя пользователя на удаленной системе (имя локального пользователя, если после имени указано ‘here’)
is		Связь между удаленным и локальным именами пользователя
to		Связь между локальным и удаленным именами пользователя
pass[word]		Пароль на удаленной системе
ssl		Соединение с сервером по указанному базоаому протоколу через защищенное SSL-соединение
sslcert		Имя файла открытого клиентского сертификата SSL
sslkey		Имя файла клиентского личного ключа SSL
sslproto		Использовать для соединения протокол ssl
folder	-r	Имя удаленного почтового ящика
smtphost	-S	Указывает серверы SMTP, на которые переправлять почту
fetchdomains		Указывает домены, на которые следует принимать почту
smtpaddress	-D	Указывает имя домена, помещаемое в строки RCPT TO
smtpname		Имя пользователя и домена, помещаемые в RCPT TO

antispam	-Z	Определяет, какие коды ответов SMTP считать за индикацию спама
mda	-m	Определяет MDA для локальной доставки
bsmtp	-o	Имя пакетного файла BSMTP
preconnect		Команда, выполняемая перед каждым соединением
postconnect		Команда, выполняемая после каждого соединения
keep	-k	Не удалять на сервере просмотренные ранее сообщения
flush	-F	Перед запросом удалить на сервере все прочтенные ранее сообщения
fetchall	-a	Принять все сообщения независимо от того, прочтенное оно или нет
rewrite		Переопределить адрес получателя для ответа (по умолчанию)
stripcr		Убрать возвраты картеки в конце каждой строки
forcecr		Поставить возвраты каретки в конце каждой строки
pass8bits		Давать команду BODY=8BITMIME на сервер ESMTP
dropstatus		Убрать строки "Status" и "X-Mozilla-Status" из входящей почты
dropdelivered		Убрать строки "Delivered-To" из входящей почты
mimedecode		Перекодировать сообщения из quoted-printable в 8-bit MIME
idle		Время простоя в ожидании новых сообщений после каждогот приема (только IMAP)
no keep	-K	Удаление прочитанных сообщений с сервера (по умолчанию).
no flush		Не удалять прочитанные сообщение перед приемом почты (по умолчанию)
no fetchall		Прием только новых сообщений (по умолчанию)
no rewrite		Не переписывать заголовки
no stripcr		Не отрезать возврат каретки (по умолчанию)
no forcecr		Не добавлять возврат каретки в конец строки (по умолчанию)
no pass8bits		Не посылать BODY=8BITMIME на сервер ESMTP listener (по умолчанию)
no dropstatus		Не удалять заголовки "Status" (по умолчанию)
no dropdelivered		Не удалять заголовки "Delivered-To" (по умолчанию)
no mimedecode		Не перекодировать quoted-printable в 8-bit MIME (по умолчанию)
no idle		Не ждать новых сообщений в конце приема почты (только IMAP)
limit	-l	Максимальный размер письма
warnings	-w	Интервал предупреждений о размере письма
batchlimit	-b	Максимальное число сообщений, пересылаемых за одно соединение
fetchlimit	-B	Максимальное количество сообщений, принимаемых за один сеанс
expunge	-e	expunge на каждом #N-м сообщении (только IMAP и POP3)
tracepolls		Добавить трассу приема в заголовок Received
properties		Игнорируется fetchmail (может использоваться в скриптах расширения)

Помните, что все опции пользователя должны следовать *после* опций сервера.

В файле .fetchmailrc строковый аргумент ‘envelope’ может быть продолжен через пробел числом. Это число, если указано, задает число пропускаемых заголовков (т.е. 1 выбирает второй заголовк указанного типа). Это иногда полезно для игнорирования паразитных строк "Received", создаваемых MDA провайдера.

Ключевые слова, не имеющие соответствия в командной строке

Опции ‘folder’ и ‘smtphost’ (в отличие от командной строки) могут принимать в качестве аргумента список имен, разделенный запятыми.

Опции, не имеющие эквивалента в командной строке: ‘via’, ‘interval’, ‘aka’, ‘is’, ‘to’, ‘dns’/‘no dns’, ‘checkalias’/‘no checkalias’, ‘password’, ‘preconnect’, ‘postconnect’, ‘localdomains’, ‘stripcr’/‘no stripcr’, ‘forcecr’/‘no forcecr’, ‘pass8bits’/‘no pass8bits’ ‘dropstatus/no dropstatus’, ‘dropdelivered/no dropdelivered’, ‘mimedecode/no mimedecode’, ‘idle/no idle’, и ‘no envelope’.

Опция ‘via’ полезна в том случае, если у вас имеется несколько настроек на один и тот же сервер. Если она присуствует, ее аргументом является имя DNS почтового сервера. Она переопределяет опцию poll, которая является просто меткой в конфигурации.

Опция ‘interval’, принимающая числовой аргумент, поволяет запрашивать сервер менее часто, чем определено основным интервалом приема. Если вы укажете ‘interval N’ , то сервер, который содержит эту опцию, будет опрашиваться каждые N интервалов приема.

Слова ‘is’ или ‘to’ ассоциируют имя локального клиента с именем пользователя на удаленном сервере (или соответствие имя_на_сервере и имя_клиента, через знак =). Если список is/to содержит в конце ‘*’, неопознанные имена пропускаются как есть.

Одиночное локальное имя может использоваться для перенаправления почты, когда ваше имя на клиенте отличается от имени на сервере. Поскольку указано только локальное имя, почта передается на этому локальному пользователю вне зависимости от содержимого заголовков Received, To, Cc и Bcc. В этом случае *fetchmail* не проверяет DNS.

Если указано более одного локального имени (или соответствия), fetchmail проверяет адреса в заголовках Received, To, Cc, and Bcc принятой почты (режим multidrop). Он ищет адреса с частью hostname, которые совпадат с вашим poll name, или вашими опциями ‘via’, ‘aka’ или ‘localdomains’, а также имена машин, найденные в DNS как псевдонимы почтового сервера.

Если *fetchmail* не может найти соответствия имени на сервере или адресов локальных доменов, почта возвращается отправителю. Однако, если указана опция "nobounce", письмо попадает постмастеру (по умолчанию это пользователь, запустивший fetchmail).

Опция ‘dns’ (обычно включенная - on) управляет способом проверки доменного почтового ящика. Она включает логику проверки адреса узла, не совпадающего с объявлениями ‘aka’ или ‘localdomains’, проверкой в DNS. Если имя пользователя сервера опознано как имя узла (hostname), его локальное соответствие добавляется в список локальных получателей.

Опция ‘checkalias’ (по умолчанию выключена - off) расширяет проверку, выполняемую опцией "dns" в режиме multidrop, помогая справиться с удаленным MTA, идентифицирующего самого себя своим каноническим именем, когда его запрашивают с использованием alias. При запросе такого сервера проверка адреса конверта не работает, и fetchmail возвращается к проверке заголовков To/Cc/Bcc. Указание этой опции заставляет fetchmail извлечь все IP-адреса, ассоциированные как с почтовым сервером, так и с именем, используемым удаленным MTA , и сравнить эти IP-адреса. Это полезно в ситациях, когда удаленный сервер часто меняет свои канонические адреса, что приводило бы к частым изменениям в конфигурационном файле. Опция "checkalias" игнорируется, если указана опция "no dns".

Опция ‘aka’ используется для доменных ящиков. Она позволяет предварительно объявить список DNS-псевдонимов сервера. Это оптимизирует скорость приема почты. Когда fetchmail обрабатывает доменный ящик, пробегая по сообщениям в поиске имен почтового сервера, предварительное объявление имен помогает сэкономить на поиске в DNS. Замечание: имена, указанные в "aka", сравниваются по суффиксам. Например, если вы указали ‘aka netaxs.com’, это будет соответствовать не только машине netaxs.com, но и любому имени машины, заканчивающемуся на ‘.netaxs.com’; т.е., например, pop3.netaxs.com и mail.netaxs.com.

Опция ‘localdomains’ позволяет объявить список доменов, которые fetchmail будет рассматривать как локальные. Если при разборке адресных строк в режиме multidrop адрес заканчивается на один из указанных доменов, этот адрес передается на MTA без изменения (т.е. преобразование в локальные адреса не производится).

Если вы используете опцию ‘localdomains’, вам может потребоваться также указать опцию ‘no envelope’, которая отключает попытку fetchmail извлечь локальные адреса из заголовков "Received" или "X-Envelope-To".

Опция **password** содержит строковый аргумент, представляющий пароль, используемый для входа на сервер.

Ключевое слово ‘preconnect’ позволяет указать внешнюю команду, запускаемую перед каждым установлением соединения с почтовым сервером. Это полезно, напрмер, если вы пытаетесь установить защищенное соединение POP с помощью ssh. Если команда возвращает ненулевой код возврата, вызов сервера не производится.

Аналогично, ключевое слово ‘postconnect’ также позволяет указать внешнюю команду, выплняемую после каждого отключения от почтового сервера.

Опция ‘forcecr’ заставляет добавлять символ CR в конец каждой строки, заканчивающейся только LF, перед отправкой на MTA. Строго говоря, в соответствии с RFC821 это надо делать, но лишь немногие MTA требуют этого, поэтому по умолчанию эта опция отключена (только для qmail это имеет значение).

Опция ‘stripcr’ определяет, надо ли убирать символы CR из принятой почты перед отправкой на MTA. По умолчанию включено.

Опция ‘pass8bits’ предназначена специально для почтовых программ Microsoft, которые тупо вставляют "Content-Transfer-Encoding: 7bit" куда ни попадя. Если эта опция отключена (по умолчанию) и присутствет такой заголовок, то fetchmail посылает BODY=7BIT на сервер ESMTP. Это может вызвать проблемы с сообщениями, использующих 8-битную кодировку (например,KOI-8), которые будут испорчены путем отсекания старшего бита. Если опция "pass8bits" включена, fetchmail посылает на SMTP команду BODY=8BITMIME. Если MTA может принимать 8-битный текст (большинство могут), то все будет правильно.

Опция ‘dropstatus’ определяет, надо ли удалять (включено по умолчанию) непустые заголовки "Status" и "X-Mozilla-Status" из принятой почты. Если их оставить, пользовательские программы чтения почты будут видеть такие сообщения как будто прочтенные на сервере. С другой стороны, это может смутить программы проверки новой почты, которые будут считать, что новых сообщений нет.

Опция ‘dropdelivered’ определяет, надо ли удалять заголовки "Delivered-To" (по умолчанию не удаляются). Эти заголовки добавляются программами qmail и Postfix для избежания зацикливания почты, но могутпомешать, если вы захотите создать "зеркальный" почтовый сервер в вашем домене. Используйте эту опцию с осторожностью.

Опция ‘mimedecode’ определяет, надо ли перекодировать кодировку сообщений "quoted-printable" в 8-битные данные. Если вы передаете почту на 8-битный ESMTP-сервер (например, sendmail), то закодированные заголовки и тело сообщения декодируются в 8-битные данные, упрощая их чтение. Если ваша программа чтения почты знает, как обрабатывать MIME-сообщения, эта опция вам не нужна.

Опция ‘idle’ используется только с серверами IMAP. Если она включена, и fetchmail обнаруживает, что сервер поддерживает IDLE, то в конце приема почты на сервер посылается команда IDLE. Это заставляет сервер держать соединение открытым и предупреждать клиента о поступлении новой почты. Если вы часто принимаете почту, IDLE может сэкономить траффик, исключая дополнительные соединения TCP/IP. С другой стороны, IDLE забирает почти все время fetchmail, поскольку соединение никогда не разрывается, пока сервер не сделает это по тайм-ауту. Кроме того, это не работает с несколькими ящиками; только с первым.

Опция ‘properties’ предусматривает механизм расширения. Она принимает строковый аргумент, который игнорируется fetchmail'ом. Этот аргумент можно использовать для хранения информации о настройке для скриптов, которым она может понадобиться. В частности, вывод опции "configdump" может сделать ассоциации свойств пользователям и использоваться в языке Python.

Прочие опции

Слова ‘here’ и ‘there’ полезны для облегчения чтения и понимания файла. Например, ‘user eric is esr’ означает, что почта удаленного пользователя eric должна быть доставлена локальному пользователю esr. Это становится еще более понятным, если записать так: ‘user eric there is esr here’.

Допустимые идентификаторы протоколов, используемые в качестве аргумента опции ‘protocol’:

auto (или AUTO)
pop2 (или POP2)
pop3 (или POP3)
sdps (или SDPS)
imap (или IMAP)
apop (или APOP)
kpop (или KPOP)

Допустимые типы аутентификации: ‘any’, ‘password’, ‘kerberos’, ’kereberos_v5’ и ‘gssapi’, ‘cram-md5’, ‘otp’, ‘ntlm’, ‘ssh‘. Тип ‘password’ определяет стандартный тип аутентификации посредством посылки пароля (пароль может быть открытым текстом или зашифрованным); ‘kerberos’ заставляет *fetchmail* попытаться получить билет Kerberos в начале каждого запроса и послать произвольную строку в качестве пароля; ‘gssapi’ заставляет использовать аутентификацию GSSAPI.

‘kpop’ означает протокол POP3 через порт 1109 с аутентификацией Kerberos V4.

Существуют также четыре глобальных оператора: ‘set logfile’ задает имя лог-файла; ‘set daemon’ задает интервал опроса серверов; ‘set postmaster’ определяет адрес, на который будет направлена почта, если не удалось определить локального пользователя. Наконец, ‘set syslog’ посылает все сообщения на syslogd.

ВЗАИМОДЕЙСТВИЕ С RFC 822

Пытаясь определить адрес отправителя сообщения, fetchmail проверяет следующие заголовки в указанном порядке:

Return-Path:
Resent-Sender: (игнорируется, если не содержит an @ или !)
Sender: (игнорируется, если не содержит an @ или !)
Resent-From:
From:
Reply-To:
Apparently-From:

Адрес отправителя используется для журнализации, и указывается в MAIL FROM при пересылке по SMTP. Такой порядок важен при анализе сообщений от списка рассылки в доменном почтовом ящике. Если локальный адрес не существует, сообщение следует вернуть не автору или в сам список, а администратору рассылки.

В режиме multidrop заголовки назначения обрабатываются следующим образом: Сначала fetchmail ищет "Received:" (или что указано в опции "envelope") для определения адреса локального получателя. Если письмо адресовано более чем одному получателю, "Received:" может не содержать информации об адресе получателя.

Затем fetchmail ищет строки Resent-To:, Resent-Cc:, и Resent-Bcc:. Если они есть, то содержат ареса окончательных получателей и имеют приоритет над To:/Cc:/Bcc:. Если строки Resent-* отсутствуют, ищутся строки To:, Cc:, Bcc: и Apparently-To:

ПРИМЕР НАСТРОЙКИ

Хотя во многих примерах присутствует объявление паролей опцией password, это сделано только в целях иллюстрации. Мы рекомендуем хранить информацию о паролях в файле `$HOME/.netrc`, откуда они могут браться не только программой fetchmail, но ftp и другими программами.

Основной формат:

```
poll SERVERNAME protocol PROTOCOL username NAME password PASSWORD
```

Пример:

```
poll pop.mail.ru protocol pop3 username "jsmith" password "secret1"
```

Или, используя сокращенную форму записи:

```
poll pop.mail.ru proto pop3 user "jsmith" password "secret1"
```

Можно перечислить несколько серверов:

```
poll pop.mail.ru proto pop3 user "jsmith" pass "secret1"
poll other.provider.com proto pop2 user "John.Smith" pass "My^Hat"
```

Здесь добавим немного удобочитаемости:

```
poll pop.mail.ru proto pop3
    user "jsmith", with password secret1, is "jsmith" here;
poll other.provider.net proto pop2:
    user "John.Smith", with password "My^Hat", is "John.Smith" here;
```

Эту версию значительно легче читать.

Если нужно включить пробелы в строку параметров, заключите эту строку в двойные кавычки. Вот так:

```
poll pop.mail.ru with proto pop3:
      user "jsmith" there has password "u can't krak this"
                  is jws here and wants mda "/bin/mail"
```

Вы можете указать настройки по умолчанию с помощью ключевого слова 'defaults' вместо 'poll'. Эта настройка будет применяться ко всем последующим запросам. Настройки по умолчанию могут быть переопределены в строке определения сервера.

```
defaults proto pop3
      user "jsmith"
poll pop.mail.ru
      pass "secret1"
poll mail.provider.net
      user "jjsmith" there has password "secret2"
```

Можно указать несколько пользователей на один сервер (полезно только если fetchmail выполняется под root). Ключевое слово user начинает определение пользователя.

```
poll pop.mail.ru proto pop3 port 3111
      user "jsmith" with pass "secret1" is "smith" here
      user jones with pass "secret2" is "jjones" here keep
```

Здесь имя локального пользователя smith связывается с именем пользователя jsmith на сервере pop.mail.ru, а локальный пользователь jjones соответствует пользователю jones на том же сервере. Причем почта для jones после приема сохраняется на сервере.

Пример конфигурации для доменного почтового ящика:

```
poll pop.provider.net:
      user maildrop with pass secret1 to golux 'hurkle'='happy' snark here
```

Тут говорится, что почтовый ящик учетной записи 'maildrop' на сервере является доменным (multidrop), а сообщения должны быть разобраны по пользователям 'golux', 'hurkle' и 'snark'. Далее указывается, что "golux" и "snark" имеют одно и то же имя на локальной системе и на сервере, но почта для "hurkle" должна быть доставлена пользователю "happy" на клиентской машине.

Вот еще один пример настройки доменного ящика:

```
poll pop.provider.net localdomains loonytoons.org toons.org:
      user maildrop with pass secret1 to * here
```

Почтовый ящик бюджета "maildrop" на сервере - это доменный ящик. Все адреса доменов loonytoons.org или toons.org (включая субдомены типа "joe@daffy.loonytoons.org") должны передаваться на локальный сервер SMTP без изменений. Берегитесь зацикливания почты!

Пример настройки с использованием опций ssh и plugin. Запросы выполняются на stdin и stdout сервера IMAP через SSH. В нашем случае аутентификация IMAP пропускается.

```
poll mailhost.net with proto imap:
       plugin "ssh %h /usr/sbin/imapd" auth ssh;
       user esr is esr here
```

ИСПОЛЬЗОВАНИЕ ДОМЕННЫХ ПОЧТОВЫХ ЯЩИКОВ

Использовать функцию множественных локальных получателей следует осторожно. Все функции multidrop не используются в ETRN или ODMR.

Учтите, что в режиме multidrop подавляется дублирование почты. Сообщение считается дубликатом, если имеет тот же идентификатор, что и предыдущее.

Заголовки и конверты

Фундаментальная проблема состотит в том, что когда ваш почтовый сервер помещает сообщения, адресованные нескольким получателям, в один почтовый ящик, возникает проблема определения адрса правильного получателя (адрес конверта, в противоположность объявленных в RFC822 заголовкам To/Cc/Bcc). Этот адрес конверта (envelope address) необходим для правильной маршрутизации почты.

Иногда *fetchmail* может определить адрес конверта. Если в качестве MTA выступает *sendmail* и почта имеет только одного получателя, MTA записывает это в виде ‘by/for’ в поле Received. Однако, это надежно не работает с другими MTA или в случае с несколькими получателями. По умолчанию fetchmail ищет адрес конверта в этих строках; вы можете восстановить это значение по умолчанию, указав опцию -E "Received" или ‘envelope Received’.

В качестве альтернативы, локальный MTA и/или почтовый сервер вставляют в каждое сообщение заголовок, соджащий копию адреса конверта. Этот заголовок (если есть) обычно ‘X-Envelope-To’. Это может быть изменено опцией -E или "envelope". Помните, что запись заголовка конверта таким способом раскрывает имена получателей (включая скрытых (Bc) получателей) для всех получателей сообщения, что может создать проблему безопасности/приватности.

Небольшой вариацией заголовка ‘X-Envelope-To’ является ‘Delivered-To’, помещяемый qmail для предотвращения зацикливания почты. Он обычно предваряет имя пользователя строкой домена пользователя. Для удаления этого префикса используйте опцию ‘qvirtual’.

Иногда, к сожалению, ни один из этих методов не срабатывает. В таком случае fetchmail откатывается к обработке содержимого To/Cc/Bcc и пытается отпределить адрес получателя. Но это не очень надежно, поскольку списки рассылки в заголовок To: часто ставят свой широковещательный адрес.

Если *fetchmail* не может определить адрес локального получателя, а подразумеваемый адрес получателя отличается от адреса пользователя,запустившего fetchmail, почта теряется. Это делает использование функций multidrop рискованным делом.

Аналогичная проблема состоит в том, что скрытая копия, информация из Bcc, передается *только* через адрес конверта (т.е. не помещается в заголовки, обрабатываемые fetchmail, если это не X-Envelope).

Правильный способ использования доменных почтовых ящиков

На клиентской стороне для администрирования списка рассылки можно использовать множетсвенные локальные имена. Допустим, ваше имя "esr", вы хотите получать собственную почту и одновременно вести список расслки, скажем, "fetchmail-friends", и хотите сохранить псевдоним на вашей клиентской машине.

На своем сервере вы можете создать псевдоним (alias) ‘fetchmail-friends’ на ‘esr’, затем в своем *.fetchmailrc* объявить ‘to esr fetchmail-friends here’. Тогда при приеме почты, включающей ‘fetchmail-friends’ в качестве локального адреса, имя списка добавится к списку получателей, который увидит SMTP. Тем не менее, локально псевдоним раскроется. Не забудьте включить "esr" в локальное раскрытие псевдонима "fetchmail-friends", иначе не увидите почту, направленную только в список рассылки. Убедитесь также, что мейлер имеет включенную опцию "me-too" (в sendmail это опция -oXm или Oxm в sendmail.cf), иначе ваше имя не удалится из раскрытия псевдонима в отправленном сообщении.

Этот трюк не идеальное решение. Вы увидите, что сообщение, пришедшее в список рассылки, будет содержать заголовок ‘X-Fetchmail-Warning’, поскольку fetchmail не смог найти подходящего локального получателя.

Плохой способ использования доменных ящиков

Нельзя смешивать доменные почтовые ящики и *fetchmail*, осблуживающий нескольких пользователей в режиме демона. Проблема опять связана со списками рассылок, которые, как правило, не указывают индивидуальных получателей. Если fetchmail не может определить адрес получателя, оно направляется пользователю, запустившему fetchmail (обычно root). Кроме того, скрытые пользователи (указанные в Bc:) никогда не получат почту.

Если вы собираетесь использовать fetchmail для приема почты для многих пользователей с единого почтового ящика по IMAP, подумайте еще раз. В этом случае проще оставить почту в очереди на сервере, а использовать ETRN или ODMR для периодического переключения потока SMTP. Или используйте UUCP.

Если вы все же *вынуждены* использовать для этого multidrop, убедитесь, что ваш почтовый сервер правильно записывает адреса конверта, чтобы fetchmail смог их определить. Иначе вы можете потерять всю вашу почту.

Ускорение проверки доменныъх ящиков

Обычно при указании нескольких пользователей fetchmail определяет адреса получателей вышеописанным способом и проверяет имя машины в DNS. Если это имя - псевдоним почтового сервера, используется карта соответствий to..here и почта доставляется локально.

Это самый безопасный и самый медленный способ. Для его ускорения можно предопределить псевдонимы почтового сервера с помощью опции "aka". Эти псевдонимы проверяются до обращения к DNS. Если вы уверены, что "aka" включает *все* псевдонимы DNS (и все записи типа MX, указывающие на него), то можете указать опцию "no dns" для полного отключения запросов к DNS.

КОДЫ ВОЗВРАТА

0 Успешно принято одно или несколько сообщений (или, с опцией -c, обнаружено)

1 Нет новой почты.

2 Ошибка при попытке открытия сокета для приема почты. Если вы не знаете, что такое сокет, не волнуйтесь на счет этого - просто считайте это "неисправимой ошибкой". Эта ошибка также возникает, если используемый протокол не прописан в файле /etc/services.

3 Ошибка аутентификации пользователя. То есть неверное имя, пароль или идентификатор APOP. Обычно свидетельствует о неправильном имени или пароле пользователя.

4 Обнаружена критическая ошибка протокола.

5 Синтаксическая ошибка в аргументах *fetchmail.*

6 Неправильные права доступа к конфигурационному файлу.

7 Сервер сообщает об ошибке, а также в случае тайм-аута при ожидании ответа от сервера.

8 Ошибка на стороне клиента. Это означает, что fetchmail обнаружил другую работающую копию fetchmail.

9 Ошибка аутентификации по причине ответа сервера "lock busy". Попробуйте еще раз через некотрое время. Это ошибка включена не во все протоколы и подерживается не всеми серверами (в этом случае возвращается код 3).

10 *fetchmail* не смог открыть порт SMTP

11 Ошибка DNS. Fetchmail обнаружил при запуске ошибку в DNS и не может продолжать работу.

12 Невозможно открыть пакетный файл BSMTP.

13 Прием почты прерван по достижении максимального количества сообщеий (см. опцию --fetchlimit).

14 Сервер занят.

23 Внутренняя ошибка. Вы должны получить более подробное разъяснение на stderr.

ФАЙЛЫ

~/.fetchmailrc

конфигурационный файл по умолчанию

~/.fetchids

файл, ассоциирующий машины с идентификаторами последних просмотренных сообщений (используется с серверами POP3, поддерживающих UIDL)

~/.fetchmail.pid

Файл локировки, предотвращающий запуск нескольких копий fetchmail (для пользователей отличных от root).

~/.netrc

Конфигурационный файл вашего FTP, в котором (если есть) ищутся ваши пароли. Если пароля не найдено, он запрашивается интерактивно.

/var/run/fetchmail.pid

Файл локировки, предотвращающий запуск нескольких копий fetchmail (для бюджета root, в системах Linux).

/etc/fetchmail.pid

Файл локировки, предотвращающий запуск нескольких копий fetchmail (для бюджета root, в системах без /var/run).

ОКРУЖЕНИЕ

Если установлена переменная окружения FETCHMAILUSER, она используется для установки имени вызывающего пользователя. В противном случае используются переменные LOGNAME или USER. Если и их нет, вызывается **getpwuid**.

Если переменная окружения FETCHMAILHOME указывает на правильный существующий каталог, то файлы .fetchmailrc, .fetchids и fetchmail.pid будут размещаться в нем (вместо домашнего каталога вызвавшего пользователя). Файл .netrc ищется в домашнем каталоге пользователя независимо от настройки FETCHMAILHOME.

СИГНАЛЫ

Если fetchmail работает в режиме демона под бюджетом root, SIGHUP пробуждает его от спячки и вызывает немедленный опрос серверов (кроме помеченных как skip).

Если fetchmail работает в режиме демона под другим бюжетом (отличном от root), то его пробуждает сигнал SIGUSR1.

Запуск fetchmail на переднем плане в то время, тогда как работает фоновый fetchmail, последний также просыпается.

ОШИБКИ И ИЗВЕСТНЫЕ ПРОБЛЕМЫ

Опции mda и plugin плохо взаимодействуют друг с другом. Чтобы получить код ошибки от MDA, в fetchmail необходимо сменить алгоритм обрабоки сигналов.

Разборщик адресов в режиме multidrop иногда сбоит на некоторых "@-адресах", которые технически верные, но странные. Особенно если используется нестандартное использование кавычек.

В сообщении с несколькими заголовками конверта fetchmail видит только последнее. Чтобы обойти это, используйте на стороне сервера фильтр, собирающий все заголовки конверта в один (procmail, mailagent, maildrop).

Использование некоторых протоколов требует отсылки паролей в открытом виде по каналам TCP/IP на почтовый сервер. Это создает риск перехвата паролей снифферами. В Linux и FreeBSD можно использовать опцию --interface, чтобы принимать почту если указанный интерфейс активен. Мы рекомендуем использовать ssh не только для шифрования вашего пароля, но и для защиты всего траффика.

Использование %F или %T в опции mda может создать брешь в безопасности,

поскольку они передают текст, который можно перехватить и подменить.

Для возврата почты и отбрасывания спама необходим открытый порт 25 на localhost.

Если вы изменили *~/.fetchmailrc* при работающем в фоновом режиме fetchmail, и нарушили синтаксис, фоновый процесс тихо умрет. К сожалению, тут ничего нельзя поделать - мы не знаем, можно ли писать в syslog.

Опция -f (чтение настроек из stdin) несовместима с опцией plugin.

Код UIDL ненадежный и может потерять свое состояние при ошибках или обрыве связи. Если это случается, переходите на IMAP.

Опция ‘principal’ используетя только в Kerberos IV, но не в V.

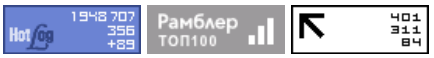
Посылайте комментарии, отчеты об ошибках и тому подобное в список рассылки "fetchmail-friends" *fetchmail-friends@lists.ccil.org*. FAQ в виде HTML на английском языке доступен по адресу http://www.tuxedo.org/~esr/fetchmail

АВТОР

Eric S. Raymond <*esr@snark.thyrsus.com*>. Слишком много людей, чтобы всех перечислить, принимало участие в разработке кода и заплат. Эта программа призвана заменить *popclient*, автор <*ceharris@mal.com*>

Русский сокращенный перевод сделан Питом>

Search...
Новые публикации
Linux Advanced Routing & Traffic Control HOWTO
Установка и настройка OpenVPN в CentOS
Установка Livestreet с нуля (Debian): nginx + mysql + php-fpm + apc + ...
Торрент клиенты под Linux
Допиливаем до ума FreeBSD или первые шаги после установки
Убийственная коллекция CSS Reset - стилей (сброса стилей браузеров, стоящих по умолчанию)
Популярные статьи
Установка и настройка эмулятора Windows - wine
ALT Linux 2.3 Compact
OpenOffice Руководство пользователя
The Linux Gamers' HOWTO
Параметры конфигурации Apache
Иерархия каталогов и файловых систем в Linux
Осваиваем Nagios
Использование GNOME
Руководство по настройке пакета веб-почты SquirrelMail
Использование KDE
Конфигурация Samba сервера
Отладка с помощью GDB
Наш баннер
Вы можете установить наш баннер на своем сайте или блоге, скопировав этот код:
<img src="http://linux.yaroslavl.ru/images/lyr-button1.gif" alt="Документация Linux по-русски" width="88" height="31" border="0">
RSS новости



Copyright © 2001-2016 Team of linux.yaroslavl.ru. E-mail: linux@yaroslavl.ru