
Qmail - Guia de consulta

Rodrigo Pereira Telles, Devel-IT [<http://www.devel.it>] <telles (at) devel (dot) it>

Copyright © 2005 por Rodrigo P. Telles.

2005-06-07

Histórico de Revisões

Revisão 1.0

2005-06-01

Release

rpt

qmail é um seguro, confiável, eficiente e simples agente de transferência de mensagens (MTA – Mail Transfer Agent). Foi desenhado para efetuar conexões pela internet entre servidores SMTP.

Hoje, é o segundo servidor SMTP mais comumente encontrado na internet e é de longe, o servidor SMTP que mais cresce na internet.

Este guia foi escrito com o intuito de auxiliar os iniciantes em qmail bem como os já experientes administradores de sistema a aproveitar ao máximo todos os recursos deste fantástico MTA.

Índice

Programas do pacote qmail	4
mailsubj	4
qmail-clean	4
qmail-inject	4
qmail-local	5
qmail-lspawn	6
qmail-pop3d	6
qmail-popup	6
qmail-qmqpc	7
qmail-qmqpd	7
qmail-qmtpd	7
qmail-queue	8
qmail-remote	11
qmail-rspawn	12
qmail-send	12
qmail-smtpd	12
qmail-start	13
sendmail	13
splogger	13
Utilitários do qmail	14
bouncesaying	15
condredirect	15
except	15
forward	16
maildirwatch	16
preline	16
qbiff	17
qreceipt	17
Utilitários de administração	17
maildir2mbox	17

maildirmake	18
qmail-getpw	18
qmail-newmrh	19
qmail-newu	19
qmail-pw2u	19
qmail-qread	21
qmail-qstat	21
qmail-showctl	21
qmail-tcpok	21
qmail-tcpto	22
tcp-env	22
Arquivos de controle/configuração	22
badmailfrom	22
bouncefrom	22
bouncehost	23
concurrencylocal	23
concurrencyremote	23
databytes	23
defaultdomain	24
defaulthost	24
doublebouncehost	24
doublebounceto	24
envnoathost	24
helohost	25
idhost	25
localiphost	25
locals	25
me	25
morercphosts	26
percenthack	26
plusdomain	26
qmqpservers	26
queuelifetime	26
rcphosts	27
smtpgreeting	27
smtproutes	27
timeoutconnect	28
timeoutremote	28
timeoutsmtpd	28
virtualdomains	28
Variáveis de Ambiente	29
DATABYTES	29
DEFAULT	29
DTLINE	29
EXT	29
EXT2	29
EXT3	29
EXT4	29
HOME	30
HOST	30
HOST2	30
HOST3	30
HOST4	30
LOCAL	30
MAILHOST/QMAILHOST	30
MAILNAME/QMAILNAME/NAME	30
MAILUSER/QMAILUSER/USER/LOGNAME	30
NEWSENDER	30

QMAILDEFAULTDOMAIN	31
QMAILDEFAULTHOST	31
QMAILIDHOST	31
QMAILINJECT	31
QMAILMFTFILE	31
QMAILPLUSDOMAIN	31
QMAILSHOST/QMAILUSER	31
RECIPIENT	32
RELAYCLIENT	32
RPLINE	32
SENDER	32
UFLINE	32
Sistema de Aliases/Forwards	32
dot-qmail	32
\$HOME/.qmail- alias	33
qmail-users - assign/cdb	35
O programa dot-forward	36
dot-forward	36
Programas do pacote fastforward	38
fastforward	38
newaliases	39
newinclude	42
printforward	42
printmaillist	43
setforward	44
setmaillist	46
O programa checkpassword	46
checkpassword	46
Sistema de entrega	48
Mailbox	48
Maildir	50
Programas do pacote ucspi-tcp	51
addr	51
argv0	52
date@	52
delcr	52
finger@	52
fixcrio	53
http@	53
mconnect	53
rblsmtpd	54
recordio	55
tcpcat	55
tcpclient	55
tcprules	57
tcprulescheck	58
tcpserver	59
who@	61
Variáveis de ambiente utilizadas por tcp-env, tcpserver e tcpclient	61
Programas do pacote daemontools	62
envdir	62
envuidgid	63
fghack	63
multilog	63
pgrphack	66
readproctitle	66
setlock	66
setuidgid	67

softlimit	67
supervise	68
svc	68
svok	69
svscan	69
svscanboot	69
svstat	70
tai64n	70
tai64nlocal	71

Programas do pacote qmail

mailsubj

mailsubj *assunto* *recipiente1* *recipiente2* ...

Recebe o corpo da mensagem via *STDIN* (entrada padrão), insere *assunto* como subject da mensagem e envia para *recipiente1* e *recipiente2*.

Ex:

```
$ cat /etc/hosts | /var/qmail/bin/mailsubj "Somente teste." foo@bar.org nospam@blackhole.org
```

qmail-clean

Remove da fila de e-mail, e-mails que não foram enviados por motivo de erro. Geralmente o e-mail é removido por motivo de tempo esgotado na tentativa de enviá-lo *N* vezes, e indisponibilidade da caixa de correio do remetente que o gerou.

qmail-inject

qmail-inject [**-nNaAhH**] [**-f** *sender*] [*recipiente1* *recipiente2* ...]

Recebe mensagens via *STDIN* (entrada padrão), adiciona informações no cabeçalho da mensagem, e invoca o programa *qmail-queue* para enviar a mensagem para um ou mais recipientes.

qmail-inject normalmente sai com código de erro 0 (zero). Se invocado incorretamente ou caso a mensagem tenha sérios problemas de sintaxe, sai com código de erro 100. Em caso de erros temporários, sai com código de erro 111.

Geralmente o programa qmail-inject é utilizado por clientes de e-mail que não suportam o protocolo SMTP para envio de e-mails, e é o substituto direto do programa *sendmail*.

Opções	Descrição
-n	Imprime a mensagem recebida via <i>STDIN</i> (entrada padrão) em <i>STDOUT</i> (saída padrão) ao invés de repassá-la ao programa <i>qmail-queue</i> .
-N	Repassa a mensagem recebida via <i>STDIN</i> para <i>qmail-queue</i> (Padrão).
-a	Envia a mensagem recebida via <i>STDIN</i> para todos os endereços de recipientes especificados via linha de comando. Ignora recipientes contidos no cabeçalho da mensagem.
-A	Envia a mensagem recebida via <i>STDIN</i> para todos os recipientes especificados via linha de comando.

Opções	Descrição
	Se nenhum endereço de recipiente for especificado via linha comando, utilizará os endereços de recipientes contidos no cabeçalho da mensagem (Padrão).
-h	Envia a mensagem recebida via <i>STDIN</i> para todos os endereços de recipientes contidos no cabeçalho da mensagem. Para mensagens que não são forwards, significa que serão utilizados os endereços de e-mail contidos nos campos <i>To</i> , <i>Cc</i> , <i>Bcc</i> e <i>Apparently-To</i> do cabeçalho da mensagem. Para mensagens que são forwards, serão utilizados os endereços de e-mail contidos nos campos <i>Resent-To</i> , <i>Resent-Cc</i> e <i>Resent-Bcc</i> do cabeçalho da mensagem. Ignora recipientes especificados via linha de comando.
-H	Envia a mensagem recebida via <i>STDIN</i> para todos os endereços de e-mail contidos no cabeçalho da mensagem e todos os recipientes especificados via linha de comando.
-f sender	Repassa a mensagem recebida via <i>STDIN</i> para o programa qmail-queue substituindo o endereço de sender da mensagem por <i>sender</i> . Substitui também o conteúdo do campo <i>Return-Path</i> do cabeçalho da mensagem e variáveis de ambiente por <i>sender</i> .
recipiente1	Endereço de e-mail do primeiro recipiente.
recipiente2	Endereço de e-mail do segundo recipiente.

qmail-local

qmail-local [-nN] *usuário diretório-home local traço extensão domínio sender defaultdelivery*

Entrega ou encaminha e-mails para contas locais.

Executa tarefas muito importantes como, detectar se o e-mail está em *loop* - quando já possui “*Delivered-To*” entre os cabeçalhos do e-mail, denunciando que o e-mail já foi entregue 1 (uma) vez para este destinatário.

O recipiente da mensagem será *local@domínio*. qmail-local criará um novo cabeçalho “*Delivered-To*” na mensagem contendo *local@domínio*.

Caso o cabeçalho “*Delivered-To: local@domain*” já exista na mensagem, qmail-local devolve a mensagem, identificando-a como em *loop* de entrega.

O sender da mensagem será *sender*, e qmail-local criará um novo cabeçalho “*Return-Path*” contendo *sender*.

diretório-home é o diretório *\$HOME* de *usuário*.

traço e *extensão* identificarão qual arquivo *.qmailtraçoextensão* será utilizado por qmail-local no momento da entrega, seguido das instruções contidas em *defaultdelivery* a serem executadas, caso o arquivo não exista.

Se a entrega da mensagem for executada com sucesso, qmail-local sai com código de erro 0 (zero).

Se a entrega falhar temporariamente, sai com código de erro 111.

Se a entrega da mensagem falhar permanentemente, qmail-local sai com código de erro diferente de 0 (zero).

Opções	Descrição
-n	Ao invés de ler e entregar a mensagem, imprime em tela a descrição das instruções de entrega.

Opções	Descrição
-N	Lê e entrega a mensagem (Padrão).

qmail-lspawn

qmail-lspawn *defaultdelivery*

Invocado pelo programa *qmail-send*, qmail-lspawn lê uma série de comandos via descriptor 0 (zero), invoca o programa *qmail-local* para a entrega da mensagem e imprime as mensagens de resultado no descriptor 1.

defaultdelivery é passado ao programa *qmail-local* como instrução padrão de entrega da mensagem.

qmail-lspawn invoca *qmail-local* de forma assíncrona, sendo assim, as mensagens de resultado podem não retornar na mesma ordem que os comandos.

Antes de invocar *qmail-local*, qmail-lspawn verifica se o endereço do recipiente não está listado no mecanismo *qmail-users*, se o endereço não estiver listado, invoca o programa *qmail-getpw*.

qmail-lspawn ignora endereços de recipientes em branco.

qmail-pop3d

qmail-pop3d *maildirname*

Permite ao usuário ler ou remover mensagens em sua caixa de e-mail no servidor POP3.

Suporta apenas formato de caixa de e-mail do tipo *Maildir* – cada e-mail é um arquivo dentro da caixa de e-mail – e geralmente utiliza a porta 110/TCP.

qmail-pop3d é geralmente invocado pelo programa *qmail-popup*, que é responsável por receber informações de usuário e senha.

maildirname indica o nome do diretório a ser utilizado por qmail-pop3d na manipulação de mensagens dentro do diretório *\$HOME* de cada usuário.

qmail-pop3d possui timeout de inatividade de 20 minutos.

qmail-pop3d suporta os comandos *UIDL*, *TOP* e *LAST*.

qmail-pop3d adiciona uma linha em branco extra a cada mensagem para corrigir alguns bugs sérios em certos clientes de e-mail.

qmail-pop3d é baseado em um programa contribuído por *Russ Nelson*.

qmail-popup

qmail-popup *nome-do-host subprograma*

Valida usuário e senha dos clientes remotos e executa *subprograma*. Suporta usuário e senha em texto plano (sem encriptação) e método de autenticação encriptada (APOP).

Geralmente *qmail-pop3d* é utilizado como *subprograma* e *nome-do-host* é utilizado em conexões APOP como uma derivação do timestamp de conexão.

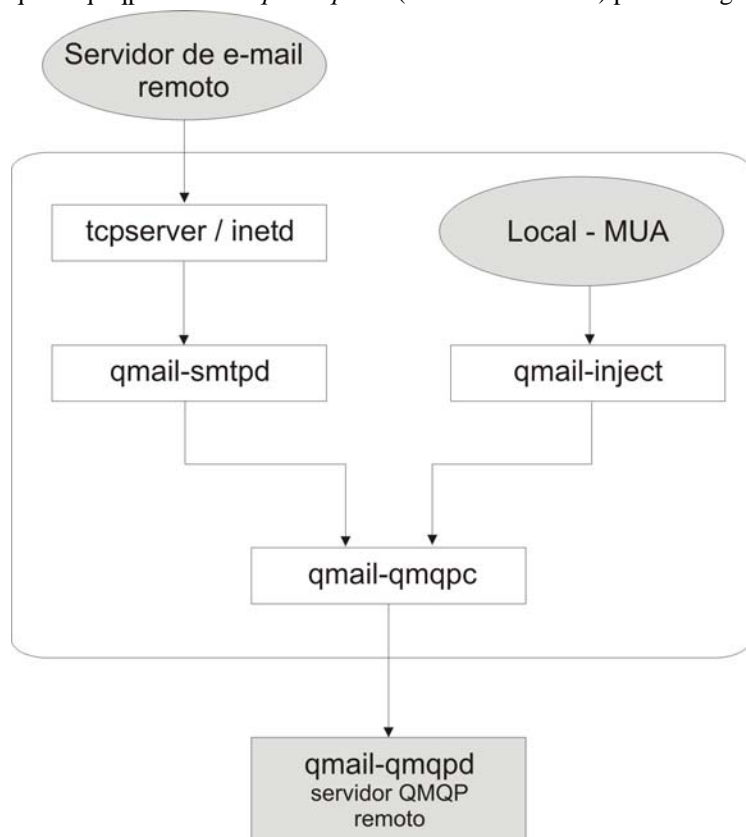
qmail-popup possui timeout de inatividade de 20 minutos.

qmail-popup é baseado em um programa contribuído por *Russ Nelson*.

qmail-qmqpc

Comunica-se com um servidor QMQP (*qmail-qmqpd*) remoto e o repassa todos os e-mails recebidos.

qmail-qmqpc substitui *qmail-queue* (via link simbólico) para entregar as mensagens recebidas ao servidor QMQP.



qmail-qmqpd

Recebe conexões do tipo QMQP de clientes *qmail-qmqpc* remotos. Uma vez que a conexão é aceita, a mensagem é entregue diretamente ao programa *qmail-queue*. Geralmente utiliza a porta 628/TCP.

Basicamente, QMQP é utilizado para enviar mensagens de um servidor qmail a outro servidor qmail de forma rápida e segura.

NOTA: QMQP (Quick Mail Queuing Protocol) é um protocolo desenvolvido por *Daniel J. Bernstein* e só pode ser utilizado entre 2 (dois) servidores qmail.

qmail-qmtpd

Recebe conexões do tipo QMTP de clientes remotos. Uma vez que a conexão é aceita, a mensagem é entregue diretamente ao programa *qmail-queue*. Geralmente utiliza a porta 209/TCP.

qmail-qmtpd suporta os mecanismos *rcpthosts*, *morecpthosts*, *RELAYCLIENTS*, *databytes* e *DATABYTES*.

NOTA: QMTP (Quick Mail Transfer Protocol) é um protocolo desenvolvido por *Daniel J. Bernstein* e tem o intuito de ser um substituto ao protocolo SMTP.

qmail-queue

Injeta mensagens na fila de e-mail para serem processadas futuramente por *qmail-send*.

É invocado pelos programas *qmail-inject*, *qmail-smtpd*, *qmail-qmtpd* e *qmail-qmqpd*.

Todo o processamento de e-mails passa por qmail-queue, tornando-o o coração do qmail.

qmail-queue sempre adiciona uma linha *Received* no início do cabeçalho da mensagem.

qmail-queue impõe algumas regras na estrutura da fila de e-mails: cada subdiretório *mess* precisa estar localizado no mesmo sistema de arquivos que o diretório *pid*; cada subdiretório *todo* precisa estar localizado no mesmo sistema de arquivos que o diretório *intd*.

qmail-queue não imprime diagnósticos em tela. Sai com código de erro 0 (zero) caso a mensagem tenha sido injetada com sucesso na fila de e-mails, e sai com código de erro entre 1 e 99 caso tenha ocorrido falha no momento de injetar a mensagem na fila.

Todos os códigos entre 11 e 40 indicam erros permanentes.

Código	Descrição
11	Endereço de e-mail muito longo.
31	O servidor de e-mail está recusando permanentemente a enviar mensagens para qualquer recipiente. (Não utilizado por qmail-queue, mas utilizado por programas que utilizam a mesma estrutura).

Todos os demais códigos indicam erros temporários.

Código	Descrição
51	Estouro de memória.
52	Tempo esgotado.
53	Erro de escrita (Ex: disco cheio).
54	Não foi possível ler a mensagem ou os cabeçalhos de entrega.
55	Não foi possível ler os arquivos de configuração (Não utilizado por qmail-queue).
56	Problemas ao tentar fazer uma conexão de rede a partir do host local (Não utilizado por qmail-queue).
61	Problemas com o diretório home do qmail.
62	Problemas com o diretório da fila de e-mail.

Código	Descrição
63	Problemas ao criar a estrutura queue/pid.
64	Problemas ao criar a estrutura queue/mess.
65	Problemas ao criar a estrutura queue/intd.
66	Problemas ao criar a estrutura queue/todo.
71	O servidor de e-mail está recusando temporariamente a enviar mensagens para qualquer recipiente. (Não utilizado por qmail-queue).
72	Tempo esgotado na conexão com o servidor de e-mail. (Não utilizado por qmail-queue).
73	Conexão com o servidor de e-mail foi rejeitada. (Não utilizado por qmail-queue).
74	Sucesso na conexão com o servidor de e-mail, mas a comunicação falhou. (Não utilizado por qmail-queue).
81	Bug interno. (Ex: falha de segmentação).
91	Erro no formato dos cabeçalhos de entrega.

Cada mensagem na fila de e-mail possui o seu próprio contador de tempo, que indica ao qmail em que momento e quantas vezes ainda deve tentar entregar a mensagem antes de devolvê-la ao seu remetente. Este tempo não é configurável, e funciona como mostrado na tabela abaixo:

Tentativa de Entrega	Segundos	Dias – HH:MM:SS
1	0	0 - 00:00:00
2	400	0 - 00:06:40
3	1600	0 - 00:26:40
4	3600	0 - 01:00:00
5	6400	0 - 01:46:40
6	10000	0 - 02:46:40
7	14400	0 - 04:00:00
8	19600	0 - 05:26:40
9	25600	0 - 07:06:40
10	32400	0 - 09:00:00

Tentativa de Entrega	Segundos	Dias – HH:MM:SS
11	40000	0 - 11:06:40
12	48400	0 - 13:26:40
13	57600	0 - 16:00:00
14	67600	0 - 18:46:40
15	78400	0 - 21:46:40
16	90000	1 - 01:00:00
17	102400	1 - 04:26:40
18	115600	1 - 08:06:40
19	129600	1 - 12:00:00
20	144400	1 - 16:06:40
21	160000	1 - 20:26:40
22	176400	2 - 01:00:00
23	193600	2 - 05:46:40
24	211600	2 - 10:46:40
25	230400	2 - 16:00:00
26	250000	2 - 21:26:40
27	270400	3 - 03:06:40
28	291600	3 - 09:00:00
29	313600	3 - 15:06:40
30	336400	3 - 21:26:40
31	360000	4 - 04:00:00
32	384400	4 - 10:46:40
33	409600	4 - 17:46:40
34	435600	5 - 01:00:00

Tentativa de Entrega	Segundos	Dias – HH:MM:SS
35	462400	5 - 08:26:40
36	490000	5 - 16:06:40
37	518400	6 - 00:00:00
38	547600	6 - 08:06:40
39	577600	6 - 16:26:40
40	608400	7 - 01:00:00

Mensagens ficam em fila, quando o qmail por algum motivo não pôde entregá-las pela primeira vez ou na tentativa anterior; tais mensagens recebem o nome de “*deferred*” (entrega falhou e foi adiada).

qmail-remote

qmail-remote *host-remoto sender recipiente [recipiente1 recipiente2 ...]*

Invocado pelo programa *qmail-rspawn*, qmail-remote recebe mensagens via entrada padrão (*STDIN*) e as entrega para recipientes localizados em servidores remotos, utilizando o protocolo SMTP.

qmail-remote envia a mensagem para *host-remoto* ou para o host listado como entrada *MX* ou *A* na zona DNS de cada domínio.

host-remoto deve ser um nome de DNS válido:

presunto.black.hole.net

ou o endereço IP do host entre colchetes:

[172.18.73.19]

O recipiente da mensagem será *recipiente* e opcionalmente *recepiente1* e *recepiente2*.

O sender da mensagem será *sender*.

qmail-remote imprime uma série de flags de recipientes, seguido de um relatório da mensagem. Cada relatório inicia com uma única letra.

qmail-remote sempre sai com código de erro 0 (zero).

Flag	Descrição
r	Relatório de recipiente: recipiente aceito.
h	Relatório de recipiente: recipiente permanentemente rejeitado.
s	Relatório de recipiente: recipiente temporariamente rejeitado.

Flag	Descrição
K	Relatório de mensagem: sucesso na entrega. Host remoto tomou a responsabilidade de entregar a mensagem para cada recipiente aceito.
Z	Relatório de mensagem: falha temporária na entrega.
D	Relatório de mensagem: falha permanente na entrega.

Depois das letras, qmail-remote imprime na tela o que ocorreu na tentativa de entrega da mensagem.

Ex:

```
$ cat /etc/hosts | /var/qmail/bin/qmail-remote foo.bar.net master@rrunner.net presunto@black-hole.net salsisha@black-hole.org
```

```
master@rrunner.net presunto@black-hole.net salsisha@black-hole.org
```

```
rrK172.18.73.19 accepted message.
```

```
Remote host said: 250 ok 1079277646 qp 1337
```

qmail-rspawn

Recebe uma série de comandos de entrega remota via descriptor 0, invoca *qmail-remote* para efetuar a entrega da mensagem e imprime os resultados no descriptor 1.

qmail-rspawn é invocado por *qmail-send*, quando este classifica o e-mail como remoto, baseando-se no endereço de recipiente.

qmail-rspawn invoca *qmail-remote* de forma assíncrona, sendo assim, as mensagens de resultado podem não retornar na mesma ordem que os comandos.

qmail-send

Entrega e-mails armazenados na fila de e-mail.

Utiliza *qmail-lspawn* para entregar mensagens à recipientes locais e *qmail-rspawn* para entregar mensagens à recipientes remotos. Se a mensagem não pôde ser entregue temporariamente para 1 (hum) ou mais recipientes, qmail-send deixa a mensagem na fila e tenta entregá-la novamente mais tarde.

qmail-send imprime registros legíveis de suas atividades no descriptor 0, escreve comandos para *qmail-lspawn*, *qmail-rspawn* e *qmail-clean* via descriptor 1 (hum), 3 (trêz) e 5 (cinco) e lê as respostas via descriptor 2 (dois), 4 (quatro) e 6 (seis).

qmail-send é responsável por evitar situações que possam causar um “beco sem saída” (deadlock).

Se qmail-send receber um sinal kill do tipo *TERM*, aguarda a finalização de todas as entregas de e-mail correntes antes de ser derrubado.

Se qmail-send receber um sinal kill do tipo *ALRM*, reagenda a entrega imediata de todas as mensagens armazenadas na fila de e-mail.

qmail-smtpd

Recebe e-mails de computadores remotos e os repassa ao programa *qmail-queue*, que irá depositar a mensagem na fila de e-mails. Utiliza o protocolo SMTP e geralmente a porta 25/TCP.

qmail-smtpd é responsável pela contagem de hops (servidores por onde a mensagem passou); rejeita qualquer e-mail com 100 (cem) ou mais entradas “*Received*” ou “*Delivered-To*” em seu cabeçalho.

qmail-smtpd suporta “*ESMTP*”, incluindo opções 8BITMIME e PIPELINING.

qmail-smtpd converte SMTP “*newlines*” em “*UNIX newlines*”, convertendo entradas “*CR LF*” em “*LF*”; qmail-smtpd retorna um erro temporário e aborta a conexão caso encontre uma entrada “*LF*” com problemas.

qmail-smtpd aceita mensagens com linhas muito longas ou caracteres não “*ASCII*”, mesmo que estas mensagens violem as especificações do protocolo SMTP.

qmail-start

qmail-start [*defaultdelivery* [*logger argumentos ...*]]

Inicia todo o sistema de entrega de e-mails

Invoca os programas *qmail-send*, *qmail-lspawn*, *qmail-rspawn* e *qmail-clean* com seus devidos UIDs (ID de usuário) e GIDs (ID de grupo). Estes serviços cooperam entre si na entrega de e-mails.

defaultdelivery é o formato padrão de entrega de mensagens que será utilizado por *qmail-lspawn*, e deve ser do tipo *Maildir* ou *Mailbox*. Se não informado, *qmail-lspawn* utilizará o formato *Mailbox* como padrão.

Se *logger* for informado, *qmail-start* repassará os logs de atividade do programa *qmail-send* para *logger*.

Variáveis de ambiente passadas à *qmail-start* são eventualmente utilizadas pelo programa *qmail-local*; tenha sempre certeza de limpar o ambiente de variáveis se executar *qmail-start* manualmente.

Ex:

```
# env - PATH="/var/qmail/bin:$PATH" qmail-start ./Maildir splogger qmail &
```

Limites de recursos e controles de *TTYs* também são repassados de *qmail-start* para *qmail-local*.

NOTA: *qmail-send* normalmente pode iniciar muitas entregas de e-mails simultâneas; para reduzir o impacto de *qmail-send* em outros programas do sistema, você pode executar *qmail-start* com prioridade baixa.

sendmail

Substitui diretamente as funcionalidades do conhecido programa sendmail. Geralmente é utilizado por clientes de e-mail que não suportam o protocolo SMTP para o envio de e-mails.

splogger

splogger [*tag* [*fac*]]

Lê e insere mensagens geradas pelo qmail no sistema de logs (syslog).

Insere *tag* (Padrão: splogger) e um timestamp numérico no início de cada mensagem.

splogger checa por *alert:* ou *warning:* no início de cada mensagem e seleciona uma prioridade syslog (*LOG_ALERT*, *LOG_WARNING* ou *LOG_INFO*) para a mensagem.

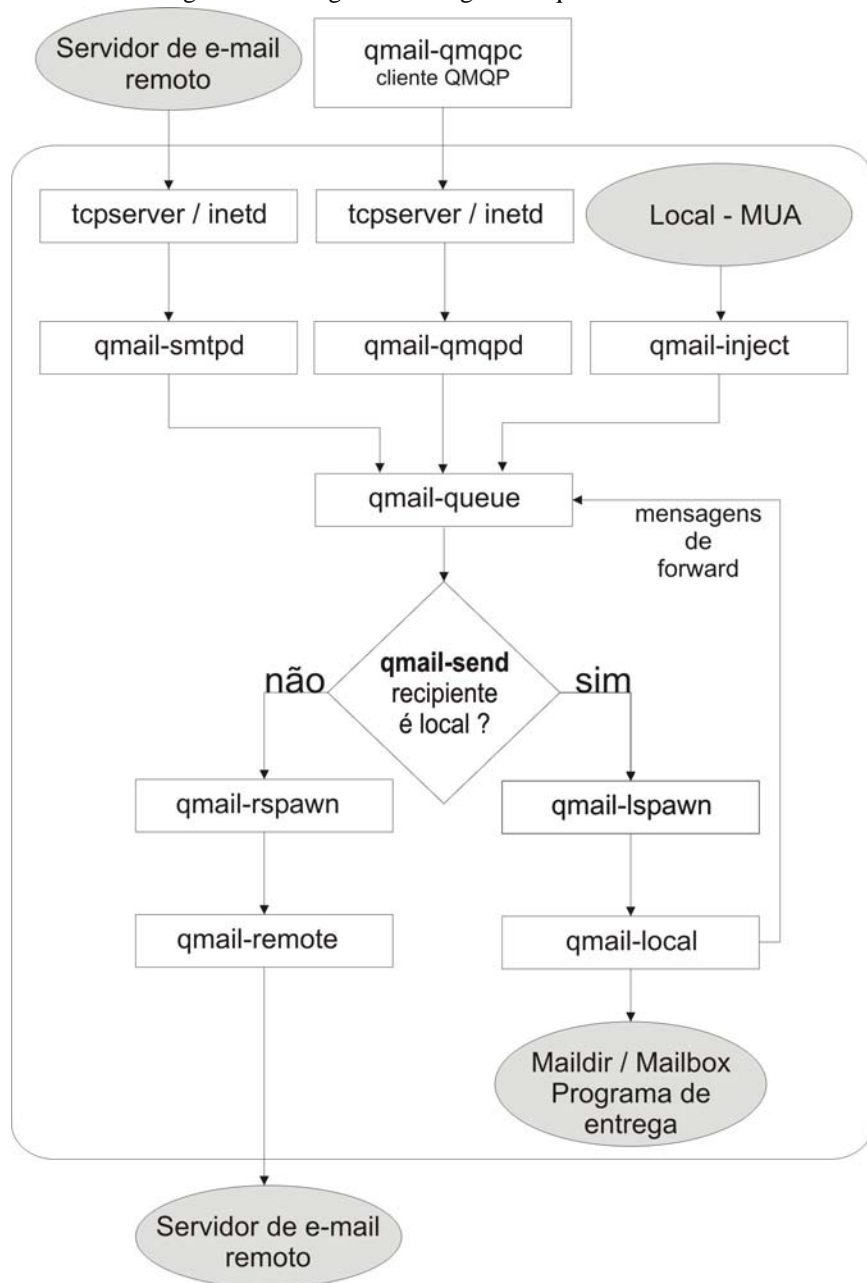
splogger loga mensagens com a facilidade *fac*. *fac* precisa ser numérico. (Padrão é 2).

splogger converte caracteres que não podem ser impressos por “?” (interrogação).

splogger não loga linhas em branco.

splogger quebra linhas que contenham mais que 800 caracteres (desde que syslog não suporte longas mensagens) e adiciona um “+” (adição) depois do timestamp para indicar uma quebra de linha.

Funcionamento geral da entrega de mensagens no qmail:



Utilitários do qmail

Estes utilitários ampliam os recursos do qmail. Basicamente, ajudam em tarefas específicas e geralmente são invoca-

dos no momento do recebimento de e-mails, ou seja, no momento anterior a entrada do e-mail na caixa do usuário. Normalmente são utilizados dentro dos arquivos *.qmail*.

bouncesaying

bouncesaying *erro* [*programa* [*argumentos* ...]]

Redireciona o e-mail para o programa *programa* (e opcionalmente os *argumentos* para serem utilizados em *programa*).

Se o código de erro retornado por *programa* for 0 (zero), o texto indicado em *erro* será enviado para o remetente, logado no sistema de logs e a mensagem não será entregue.

Se *programa* retornar qualquer outro código de erro ou não existir, bouncesaying sai com código de erro 0 (zero) e continua o processo.

Se não for informado nenhum *programa*, bouncesaying sempre retornará a mensagem indicada em *erro*:

bouncesaying 'This address no longer accepts mail.'

condredirect

condredirect *novoendereço* *programa* [*argumentos*]

Redireciona o e-mail para o programa *programa* (e opcionalmente os *argumentos* para serem utilizados em *programa*).

Se o código de erro retornado por *programa* for 0 (zero), condredirect redireciona o e-mail para *novoendereço* e sai com código de erro 99 (noventa e nove), ignorando qualquer outro comando.

Se o código de erro retornado por *programa* for 111 (cento e onze), condredirect sai com código de erro 111, o e-mail volta para a fila de e-mails e a entrega será tentada novamente mais tarde.

Se *programa* retornar qualquer outro código de erro ou não existir, condredirect sai retornando código de erro 0 (zero) e continua o processo de entrega da mensagem.

OBS: Se habilitar condredirect em um arquivo *.qmail*, não se esqueça de adicionar uma linha especificando a entrega na sua caixa de e-mail.

Ex: conteúdo do arquivo *\$HOME/.qmail*

| **condredirect** *telles@newdomain.com* *\$HOME/meuscript.sh*

except

except *programa* [*argumentos*]

Altera o código de retorno de erro de um programa.

Se *programa* retornar o código de erro 0 (zero), except sai com código de erro 100 (cem).

Se *programa* sair com código de erro 111 (cento e onze), except também sai com código de erro 111.

Qualquer outro código de erro retornado por *programa* faz com que except saia com código de erro 0 (zero).

except é normalmente utilizado em conjunto com *bouncesaying* e *condredirect*.

Ex: conteúdo do arquivo *\$HOME/.qmail*

| **bouncelaying** 'Apenas p/ o assunto qmail !' except grep -qi 'Sobre qmail'

./Maildir/

O exemplo acima, faz com que *bouncelaying* retorne o e-mail para o destinatário caso o e-mail não contenha as palavras 'Sobre qmail' no cabeçalho do e-mail ou em seu corpo.

forward

forward *recipiente1* *recipiente2* ...

Redireciona novos e-mails para um ou mais endereços de recipientes.

Possui o mesmo efeito que inserir cada endereço dentro de um arquivo *.qmail*, mas com a vantagem de poder construir o endereço de e-mail do recipiente no momento da entrega.

Ex:

| **forward** \${LOCAL/rodrigo/rodrigo-lixo}@\${HOST}

No exemplo anterior, quando utilizado no arquivo *.qmail* dentro do diretório *\$HOME* do usuário *rodrigo*, **forward** encaminhará todas as novas mensagens para a conta *rodrigo-lixo* do mesmo domínio que *rodrigo*.

maildirwatch

Monitora o diretório de e-mail do usuário e reporta quando novas mensagens são recebidas, exibindo um pequeno sumário com as novas mensagens.

maildirwatch imprime na tela, um sumário de novas mensagens 2 (duas) vezes por minuto.

maildirwatch utiliza a variável de ambiente *\$MAILDIR* para identificar o diretório a ser monitorado.

Foi desenvolvido para ser executado em um terminal compatível com *VT100*; limpa a tela antes da impressão de cada sumário.

preline

preline [-dfr] *programa* [*argumentos*]

Sujeita cada nova mensagem através de *programa*.

Por padrão, preline adiciona 3 (três) campos no cabeçalho do e-mail.

Os campos *From:*, *Return-Path:* e *Delivered-To:* são adicionados ao cabeçalho do e-mail (caso já não existam) com seus respectivos valores.

Algumas opções podem ser utilizadas para modificar a ação padrão de preline.

Opção	Descrição
-d	Não insere o campo <i>Delivered-To:</i> . Utilize esta opção quando o recipiente da mensagem de entrada está atualmente sob controle remoto, mas foi enviada aqui através de <i>control/virtualdomains</i> para roteamento manual.

Opção	Descrição
-f	Não insere o campo <i>From</i> ..
-r	Não insere o campo <i>Return-Path</i> ..

qbiff

Imprime uma mensagem no terminal do usuário quando um novo e-mail chegar em sua caixa de e-mails.

Para que esta notificação funcione, é necessário que o programa qbiff esteja devidamente configurado no arquivo *\$HOME/.qmail* do usuário.

Ex:

./Maildir/

| qbiff

OBS: É necessário habilitar o recebimento de mensagens no terminal do usuário via 'biff y'.

greceipt

greceipt *seu-email*

Responde à mensagens que solicitam uma confirmação de recebimento.

Se *seu-email* estiver contido no campo *Notice-Requested-Upon-Delivery-To*: do cabeçalho do e-mail, greceipt enviará uma notificação de recebimento ao remetente.

OBS: Se habilitar greceipt em um arquivo .qmail, não se esqueça de adicionar uma linha especificando a entrega na sua caixa de e-mail.

Ex:

./Maildir/

| greceipt telles@domain.org

A mensagem que o remetente da mensagem receberá, será algo como:

From: DELIVERY NOTICE SYSTEM <endereco-remetente>

To: <endereco-remetente>

Subject: success notice

Hi! This is the greceipt program. Your message was delivered to the following address: telles@domain.org. Thanks for asking.

Utilitários de administração

maildir2mbox

Converte caixas de correio no formato *Maildir* para o formato *mbox* (Mailbox).

Requer 3 (três) variáveis de ambiente setadas:

Variável	Descrição
\$MAIL	Localização do Mailbox do usuário a ser criado.
\$MAILTMP	Localização do arquivo temporário de transição a ser utilizado.
\$MAILDIR	Localização do Maildir do usuário a ser convertido.

\$MAIL e *\$MAILTMP* precisam estar no mesmo sistema de arquivos.

maildir2mbox não remove a mensagem de *\$MAILDIR* enquanto não escreve-la completamente dentro de *\$MAIL*.

maildir2mbox trava o arquivo *\$MAIL* para protegê-lo de acesso simultâneo por um cliente de e-mail. Este sistema de trava não protege *\$MAIL* de acesso simultâneo por outro processo *maildir2mbox*. Execute apenas uma instância de *maildir2mbox* por vez.

maildirmake

maildirmake *Maildir*

Cria o diretório *Maildir* utilizando o formato de caixa de correio *Maildir*.

maildirmake criará a estrutura *new*, *cur* e *tmp* dentro de *Maildir*, aonde *Maildir* deve ser criado dentro do diretório *\$HOME* do usuário e usuário deve ser o dono de *Maildir* e todos os seus subdiretórios.

qmail-getpw

qmail-getpw *local*

Procura o usuário que controla o endereço *local*. Retorna *usuário*, *Unix userid*, *Unix groupid*, *\$HOME*, *traço* e *extensão*.

Quando *local* é igual a *usuário*, *traço* e *extensão* retornam vazios.

Quando *local* é igual a *usuário-algumacoisa*, *traço* será “-” e *extensão* será “*algumacoisa*”.

Aonde, mensagens para *local* serão gerenciadas por *\$HOME/.qmail-traçoextensão*.

Se *qmail-getpw* não encontrar o usuário que controla *local*, o usuário *alias* entra em cena e toma o controle da mensagem. Neste caso, *traço* será “-” e *ext* será “*local*”.

Se *qmail-getpw* encontrar problemas na execução, sai com código de erro diferente de 0 (zero) e não imprime nada em tela.

qmail-getpw considera como conta local todos os usuários do arquivo */etc/passwd* se:

- Não possuir UID 0 (zero).

- O diretório *\$HOME* do usuário existir e for visível por *qmail-getpw*.
- O usuário é o dono de seu diretório *\$HOME*.

qmail-getpw ignora nomes de usuários que contenham letras em maiúsculas.

qmail-getpw assume que todos os nomes de usuários possuem menos de 32 caracteres.

OBS: O mecanismo *qmail-users* cancela as decisões tomadas por *qmail-getpw*, é confiável, altamente configurável e muito mais rápido que *qmail-getpw*.

qmail-newmrh

Cria o arquivo base de dados *control/morecphosts.cdb* em formato binário a partir do arquivo *control/morecphosts*, que será utilizado pelo programa *qmail-smtpd* de forma mais eficiente.

Caso exista algum problema com o arquivo *control/morecphosts*, *qmail-newmrh* exibe uma mensagem de erro e não gera o novo arquivo *control/morecphosts.cdb*.

qmail-newmrh assegura que *control/morecphosts.cdb* seja atualizado atômicamente (atualizações simultâneas), desta forma, *qmail-smtpd* não precisa aguardar *qmail-newmrh* liberar o arquivo para acessá-lo de forma segura.

qmail-newmrh não tenta proteger *control/morecphosts.cdb* contra atualizações simultâneas.

O binário *control/morecphosts.cdb* é portátil entre máquinas diferentes.

qmail-newu

Cria o arquivo database */var/qmail/users/cdb* em formato binário a partir do arquivo */var/qmail/users/assign*, que será utilizado por *qmail-lspawn* de forma mais eficiente.

Caso exista algum problema na leitura do arquivo *users/assign*, *qmail-newu* exibe uma mensagem de erro e não gera o novo arquivo *users/cdb*.

qmail-newu assegura que *users/cdb* seja atualizado atômicamente (atualizações simultâneas), desta forma, *qmail-lspawn* não precisa aguardar *qmail-newu* liberar o arquivo para acessá-lo de forma segura.

qmail-newu não tenta proteger *users/cdb* contra atualizações simultâneas.

O binário *users/cdb* é portátil entre máquinas diferentes.

qmail-pw2u

qmail-pw2u [*-/ohHuUC*] [*-c char*]

Lê é um arquivo de senhas no padrão V7 da entrada padrão e imprime na saída padrão no formato *qmail-users*.

Geralmente a saída produzida pelo comando *qmail-pw2u* deve ser gravada no arquivo *users/assign*.

Cada linha de um arquivo de senhas no formato V7 deve ter a forma:

usuário:senha:uid:gid:gecos:home:shell

Aonde, *usuário* é o nome da conta local, *uid* e *gid* são o User ID e Group ID de *usuário*, *home* é o diretório *\$HOME*

de *usuário*. *senha*, *gecos* e *shell* são ignorados por *qmail-pw2u*.

qmail-pw2u ignora o usuário se:

- Possuir UID 0 (zero).
- Diretório *\$HOME* não existir.
- O usuário não for o dono de seu diretório *\$HOME*.
- O nome de usuário possuir letras em maiúsculo.

Se o conteúdo impresso por *qmail-pw2u* for gravado no arquivo */var/qmail/users/assign* e for executado o comando *qmail-newu*, *qmail-lspawn* irá obedecer todas as instruções impressas por *qmail-pw2u*.

OBS: Após alterar qualquer usuário, uid, gid ou diretório *\$HOME* dentro do arquivo de senhas, é necessário executar *qmail-pw2u* e *qmail-newu* novamente, para que *qmail-lspawn* obedeça às alterações.

As ações padrão de *qmail-pw2u* podem ser alteradas, utilizando-se os seguintes arquivos dentro do diretório */var/qmail/users*:

- *include*

Lista de usuários permitidos, um por linha.

Se *include* existir, e o usuário não estiver listado neste arquivo, será ignorado.

- *exclude*

Lista de usuários que devem ser ignorados, um por linha.

Se *exclude* existir, e o usuário estiver listado neste arquivo, será ignorado.

- *mailnames*

Nomes substitutos para nomes de usuários, cada linha deve ter a forma:

usuario:mailname1:mailname12Aonde mensagens para os endereços de e-mail *mailname1*, *mailname1-algumacoisa*, *mailname2* serão entregues para *usuario*.

OBS: Mensagens para os endereços *usuário* e *usuário-algumacoisa* não serão entregues para *usuário*, a menos que estejam listados como um dos *mailnames*.

- *subusers*

Endereços extra para nomes de usuários, cada linha deve ter a forma:

sub:usuario:extra:*sub* será gerenciado por *\$HOME/.qmail-extra*, aonde *\$HOME* será o diretório *\$HOME* de *usuário.sub*. *sub-algumacoisa* será gerenciado por *\$HOME/.qmail-extra-algumacoisa*.

- append

Tarefas extras, que serão impressas na tela no final do processamento de *qmail-pw2u*.

Pode-se ainda utilizar as opções:

Opção	Descrição
-o	Ignora usuário se seu diretório <i>\$HOME</i> não existir (ou não seja visível por <i>qmail-pw2u</i>) ou caso não seja o dono de seu diretório <i>\$HOME</i> (Padrão).
-h	Pára o processamento se o diretório <i>\$HOME</i> do usuário não existir (apropriado se cada usuário deve ter seu diretório <i>\$HOME</i>) ou caso não seja o dono de seu diretório <i>\$HOME</i> .
-H	Não checa se o usuário é dono de seu diretório <i>\$HOME</i> .
-u	Permite letras em maiúsculo no nome do usuário.
-U	Ignora usuário, se houver alguma letra em maiúsculo no nome do usuário (Padrão).
-c <i>char</i>	Utiliza <i>char</i> como delimitador de extensão de usuário no lugar de “-”
-C	Desabilita o mecanismo de extensão de usuários.
-/	Utiliza <i>\$HOME/.qmail-/...</i> ao invés de <i>\$HOME/.qmail-...</i>

qmail-qread

Gera e mostra um relatório detalhado das mensagens que estão aguardando na fila do qmail. Este relatório mostra a data que a mensagem entrou em fila, tamanho em bytes, remetente e todos os recipientes que devem receber a mensagem.

Deve ser executado pelo usuário root ou pelo usuário qmails membro do grupo qmail.

qmail-qstat

Gera e mostra um relatório com o número de mensagens que estão aguardando na fila do qmail para serem processadas e entregues.

Deve ser executado pelo usuário root ou por algum usuário do grupo qmail.

qmail-showctl

Analisa os arquivos de configuração do qmail e mostra sua configuração corrente de forma detalhada.

qmail-tcpok

Limpa a tabela de timeout TCP do programa *qmail-remote*, fazendo com que *qmail-remote* não utilize nenhuma referência de tempo para os endereços que falharam.

Deve ser executado pelo usuário *root* ou pelo usuário *qmailr* membro do grupo *qmail*.

qmail-tcpto

Gera e mostra um relatório com os timeouts da tabela TCP do programa *qmail-remote*.

Se o programa *qmail-remote* falhar em uma tentativa de conexão SMTP, o IP do host remoto é gravado nesta tabela. Se a conexão SMTP para este host falhar uma segunda vez (depois de no mínimo 2 minutos de conexões sem sucesso), esta entrada é marcada na tabela e o qmail não tentará entregar esta mensagem por 1 (uma) hora.

Deve ser executado pelo usuário *root* ou pelo usuário *qmailr* membro do grupo *qmail*.

tcp-env

tcp-env [-rR] [-t n] *programa* [*argumentos*]

Tenta obter informações sobre conexões TCP para setar uma série de variáveis de ambiente que serão utilizadas por *programa* (e opcionalmente seus *argumentos*).

A entrada de dados para o programa tcp-env deve ser uma conexão do tipo TCP.

tcp-env é geralmente utilizado em conjunto com o programa *inetd*.

Opção	Descrição
-r	Tenta obter informação do host remoto e setar a variável <i>\$TCPREMOTEINFO</i> (Padrão).
-R	Não tenta obter <i>\$TCPREMOTEINFO</i> do host remoto.
-t n	Tempo máximo em segundos que tentará obter <i>\$TCPREMOTEINFO</i> de uma conexão TCP antes de desistir. O padrão é 30 segundos.

Arquivos de controle/configuração

badmailfrom

Listagem de remetentes que serão proibidos de enviar e-mails para este servidor ou para servidores remotos.

Ex:

null@spammer.com

@spammers.com

Utilizado por: *qmail-smtpd*.

bouncefrom

Define o nome de usuário (sender) que será utilizado em mensagens de erro que serão retornadas aos seus respectivos recipientes.

Este usuário não representa um usuário local do sistema.

Padrão: *MAILER-DAEMON*.

Utilizado por: *qmail-send*.

bouncehost

Define o nome de host que será utilizado em mensagens de erro que serão retornadas aos seus respectivos recipientes.

Se a mensagem não puder ser entregue permanentemente, *qmail-send* envia uma única mensagem de bounce para o *sender*.

O *sender* terá o formato *bouncefrom@bouncehost*, caso contrário será vazio.

Padrão: *me*.

Utilizado por: *qmail-send*.

concurrencylocal

Define o número máximo de entregas locais simultâneas que o qmail pode gerenciar.

Padrão: *10* - podendo se estender a *120* (limite máximo definido na compilação do qmail). Se for 0 (zero), as entregas locais serão colocadas em espera.

Utilizado por: *qmail-send*.

concurrencyremote

Define o número máximo de entregas remotas simultâneas que o qmail pode gerenciar.

Padrão: *20* - podendo se estender a *120*. (limite máximo definido na compilação do qmail). Se for 0 (zero), as entregas remotas serão colocadas em espera.

Utilizado por: *qmail-send*.

databytes

Define o tamanho máximo em bytes que um e-mail pode ter ou 0 (zero) para sem limite. Para definir o tamanho máximo de e-mails em 10MB, utilize:

10485760

Se a mensagem exceder este limite, *qmail-smtpd* retorna um código de erro permanente ao cliente.

Se o disco local estiver cheio ou *qmail-smtpd* alcançar um limite de recursos, *qmail-smtpd* retorna um código de erro temporário ao cliente.

Padrão: *0*.

Utilizado por: *qmail-smtpd*.

defaultdomain

Domínio padrão que será utilizado em endereços de e-mail que não possuam a forma *usuario@nomedohost.dominio*.

Por padrão, qmail-inject adiciona o conteúdo de defaultdomain a qualquer nome de host que não possua o domínio; também válido para *defaulthost*.

Exceções: veja *plusdomain*.

Padrão: *me*.

Utilizado por: *qmail-inject*.

defaulthost

Nome de host padrão que será utilizado em endereços de e-mail que não possuam a forma *usuario@nomedohost.dominio*.

Por padrão, qmail-inject adiciona o conteúdo de defaulthost a qualquer endereço de e-mail que não possua o nome do host.

defaulthost não precisa ser o nome corrente do servidor.

Padrão: *me*.

Utilizado por: *qmail-inject*.

doublebouncehost

Especifica o nome de host que será utilizado no *sender* de mensagens de erro que foram enviadas e retornaram com erro.

Padrão: *me*.

Utilizado por: *qmail-send*.

doublebounceto

Especifica o nome de usuário que será utilizado para receber mensagens de erro que foram enviadas e retornaram com erro.

qmail-send envia mensagens desse tipo para *doublebounceto@doublebouncehosts*; se a mensagem retornar com erro, qmail exclui a mensagem.

Padrão: *postmaster*.

Utilizado por: *qmail-send*.

envnoathost

Define o domínio padrão a ser utilizado para recipientes que não possuem “@” em seu endereço de e-mail.

Se *qmail-send* encontrar um recipiente sem “@” em seu endereço de e-mail, adiciona *@envnoathost* ao endereço de e-mail.

Padrão: *me*.

Utilizado por: *qmail-send*.

helohost

Nome corrente do host, para ser utilizado por *qmail-remote* em sessões SMTP somente para identificar-se ao servidor remoto.

Padrão: *me*.

Utilizado por: *qmail-remote*.

idhost

Nome do host que será utilizado no cabeçalho *Message-ID* em e-mails.

idhost não precisa ser o nome corrente do servidor.

Padrão: *me*.

Utilizado por: *qmail-inject*.

localiphost

Altera o nome da máquina associado ao IP local.

qmail-smtpd é responsável por detectar endereços IP no formato *a.b.c.d*. Quando *qmail-smtpd* detecta um endereço de e-mail no formato *box@[a.b.c.d]* aonde *[a.b.c.d]* é um endereço IP local, altera pelo conteúdo de localiphost.

Isso é feito antes do processamento de *rcpthosts*.

Ex:

qmailserver.domain.net

Padrão: *me*.

Utilizado por: *qmail-smtpd*.

locals

Lista de domínios considerados locais pelo qmail, um por linha.

O e-mail *usuario@dominio.com* será considerado local se *dominio.com* estiver listado em locals.

Padrão: *me*.

Utilizado por: *qmail-send*.

me

Especifica o nome do host do servidor de e-mail local.

É o arquivo de controle mais importante do qmail, pois seu conteúdo é utilizado como padrão por muitos outros arquivos, caso não estejam presentes. Sem este arquivo o qmail não funciona.

Utilizado por: *qmail-send*.

morercpthosts

Listagem de hosts e domínios adicionais que o qmail reconhecerá como locais e aceitará mensagens para seus destinatários.

Utilizando *qmail-newmrh*, será gerado um arquivo base de dados no formato binário (*control/morercpthosts.cdb*), que será utilizado por *qmail-smtpd* para consultas mais rápidas e eficientes. É necessário executar *qmail-newmrh* toda vez que o conteúdo do arquivo *morercpthosts* for alterado, para que as alterações sejam efetivadas.

Dica para grande quantidade de domínios: mantenha os 50 domínios mais acessados no arquivo *rcpthosts* e o restante em *morercpthosts*.

Utilizado por: *qmail-smtpd*.

percenthack

Especifica a lista de domínios aonde será checado a presença de “%” (sinal de percentagem) no nome do usuário.

Nomes de usuários do tipo *usuario%empresa.com@dominio.com* serão convertidos para *usuario@empresa.com*.

qmail-send processa *percenthack* antes de *locals*.

Utilizado por: *qmail-send*.

plusdomain

Extensão que *qmail-inject* adicionará ao final de domínios que terminam com o sinal “+” (adição); também válido para *defaulthost*, se este terminar com um sinal “+”.

Se o nome do host não possuir a forma *nomedohost.dominio* mas terminar com o sinal “+”, *qmail-inject* utilizará *plusdomain* e não *defaultdomain*.

Ex:

com.br

Padrão: *me*.

Utilizado por: *qmail-inject*.

qmqpservers

Especifica os endereços de servidores QMQP que serão utilizados pelo programa *qmail-qmqpc*.

Utilizado por: *qmail-qmqpc*.

queuelifetime

Define o número máximo em segundos que uma mensagem pode ficar na fila do qmail.

Depois desse tempo, *qmail-send* tentará entregar a mensagem mais uma vez e tratará qualquer outro erro, como um erro permanente.

Padrão: *604800* (uma semana).

Utilizado por: *qmail-send*.

rcpthosts

Listagem de hosts e domínios que serão reconhecidos pelo qmail como locais e serão aceitas mensagens para seus recipientes. Caso não exista, *qmail-smtpd* aceitará e-mails para qualquer host ou domínio.

A documentação do qmail recomenda adicionar não mais que 50 hosts ou domínios em *rcpthosts*.

Domínios adicionais devem ser adicionados em *morercpthosts*.

Exceção: Se a variável de ambiente *\$RELAYCLIENT* estiver setada, *qmail-smtpd* ignora *rcpthosts* e adiciona o conteúdo de *\$RELAYCLIENT* ao endereço do recipiente.

Ex:

qmailserver.domain.net

domain.com.br

Utilizado por: *qmail-smtpd*.

smtpgreeting

Especifica a mensagem de boas vindas que será exibida no banner apresentado por *qmail-smtpd*.

Padrão: *me*.

Utilizado por: *qmail-smtpd*.

smtproutes

Especifica rotas SMTP estáticas para entrega de e-mails em hosts remotos.

Cada linha deve possuir o formato:

dominio:servidor-remoto

ou

dominio:servidor-remoto:porta-tcp

Ex:

dominio.com:[172.11.10.12]

dominio.net:smtp2.dominio.net

dominio.org:mail.dominio.org:26

Aonde, todos e-mails enviados para *dominio.com* serão entregues no host *172.11.10.12*, e-mails enviados para *dominio.net* serão entregues no host *smtp2.dominio.net* e e-mails enviados para *dominio.org* serão entregues no host *mail.dominio.org* na porta *26/TCP*.

qmail-remote não tentará obter a entrada *MX* ou *A* via *DNS* para *dominio.com*, *dominio.net* e *dominio.org*.

Se *servidor-remoto* não for especificado, *qmail-remote* atuará de forma padrão, tentando obter a entrada *MX* para *dominio* através de *DNS*.

Ex:

.co.uk:

:master.co.uk

Aonde, mensagens enviadas para domínios que terminam em “.co.uk” (mas não para co.uk) serão roteados através de sua entrada *MX*; qualquer outro endereço será roteado artificialmente para o host *master.co.uk*.

Utilizado por: *qmail-remote*.

timeoutconnect

Define o número máximo em segundos que *qmail-remote* aguardará o servidor SMTP remoto aceitar a conexão SMTP antes de desconectar.

Padrão: *60 segundos*.

Utilizado por: *qmail-remote*.

timeoutremote

Define o número máximo de segundos que *qmail-remote* aguardará por uma resposta do servidor SMTP remoto (depois da conexão ser aceita) antes de desconectar.

Padrão: *1200 segundos* (20 minutos).

Utilizado por: *qmail-remote*.

timeoutsmtpd

Especifica o número máximo em segundos que *qmail-smtpd* irá aguardar para receber dados do cliente SMTP remoto.

Se nenhum dado for recebido por *qmail-smtpd* ao término de timeoutsmtpd, a conexão SMTP remota será terminada.

Padrão: *1200 segundos* (20 minutos).

Utilizado por: *qmail-smtpd*.

virtualdomains

Lista de usuários ou domínios que serão tratados pelo qmail como virtuais.

Um usuário virtual possui a forma *usuario@dominio.com:prepend*.

Quando *qmail-send* receber um e-mail para *usuario@dominio*, converterá para *prepend-usuario@dominio* e aceitará este usuário como local.

Domínios virtuais possuem a forma *dominio.com:prepend*, que será convertido para *prepend-dominio.com* para todos os usuário deste domínio.

É permitido a utilização de curingas.

Ex:

.dominio.com:telles

:alias-pegatudo

Aonde, mensagens enviadas para qualquer usuário do domínio *dominio.com* serão processadas como *telles-usuario@dominio.com* e *alias-pegatudo* receberá todas as outras mensagens que não possuírem como destinatário um usuário do domínio *dominio.com*.

Exceção: O domínio não será tratado como virtual se o *prepend* for vazio.

qmail-send processa virtualdomains depois de *locals*; se o domínio estiver listado em *locals*, não será checado em virtualdomains.

Utilizado por: *qmail-send*.

Variáveis de Ambiente

DATABYTES

Seu conteúdo tem precedência sobre o valor definido no arquivo de controle *databytes*.

DEFAULT

Contêm a parte correspondente a “default” de arquivos *.qmail-...*

Ex: Mensagem enviada para *telles-amigos*.

Se o e-mail alcançar o arquivo *.qmail-default* no diretório *\$HOME* do usuário *telles*, a variável de ambiente *DEFAULT* assumirá o valor “*amigos*”.

DTLINE

Linha completa de *Delivered-To*: utilizada no cabeçalho do e-mail.

EXT

Extensão do endereço de e-mail.

Ex: *telles-amigos-sp-local-area*.

EXT assumirá o valor “*amigos*”.

EXT2

Parte do endereço de e-mail a partir do segundo “-” (traço).

EXT2 assumirá o valor “*sp*”.

EXT3

Parte do endereço de e-mail a partir do terceiro “-” (traço).

EXT3 assumirá o valor “*local*”.

EXT4

Parte do endereço de e-mail a partir do quarto “-” (traço).

EXT4 assumirá o valor “*area*”.

HOME

Diretório *\$HOME* do usuário local.

HOST

Domínio utilizado no e-mail.

Ex: *sub.dominio.com.br*.

HOST2

Parte do domínio que precede o último “.” (ponto).

Ex: *sub.dominio.com*.

HOST3

Parte do domínio que precede o penúltimo “.” (ponto).

Ex: *sub.domínio*.

HOST4

Parte do domínio que precede o ante-penúltimo “.” (ponto).

Ex: *sub*.

LOCAL

Nome local do e-mail.

MAILHOST/QMAILHOST

Seu conteúdo tem precedência sobre o nome do host definido no arquivo de controle *defaulthost*.

MAILNAME/QMAILNAME/NAME

Nome pessoal utilizado no campo *From:* do cabeçalho do e-mail.

MAILUSER/QMAILUSER/USER/LOGNAME

Nome do usuário utilizado no campo *From:* do cabeçalho do e-mail.

NEWSENDER

Armazena o endereço de e-mail do sender utilizado para redirecionamento.

QMAILDEFAULTDOMAIN

Seu conteúdo tem precedência sobre o valor definido no arquivo de controle *defaultdomain*.

QMAILDEFAULTHOST

Seu conteúdo tem precedência sobre o valor definido no arquivo de controle *defaulthost*.

QMAILIDHOST

Seu conteúdo tem precedência sobre o valor definido no arquivo de controle *idhost*.

QMAILINJECT

Especifica opções adicionais ao programa *qmail-inject*.

Opção	Descrição
c	Utiliza o formato endereço-comentário para o campo <i>From</i> . Normalmente, <i>qmail-inject</i> utiliza o formato nome-endereço.
s	Não verifica nenhum campo <i>Return-Path</i> em mensagens de entrada. Normalmente, se <i>Return-Path</i> é especificado, este é setado como endereço de sender e sobrescreve o valor de todas as variáveis de ambiente. <i>Return-Path</i> é deletado de qualquer forma.
f	Deleta qualquer campo <i>From</i> em mensagens de entrada. Normalmente, se <i>From</i> é especificado, este sobrescreve o valor do campo <i>From</i> criado por <i>qmail-inject</i> .
i	Deleta qualquer campo <i>Message-ID</i> em mensagens de entrada. Normalmente, se <i>Message-ID</i> é especificado, este sobrescreve o valor do campo <i>Message-ID</i> criado por <i>qmail-inject</i> .
r	Utiliza <i>Return-Path</i> por usuário (VERP - Variable Envelope Return Paths). Isto faz com que <i>qmail-inject</i> adicione cada endereço de recipiente no sender da cópia da mensagem que está indo para este usuário.
m	Utiliza <i>Return-Path</i> por mensagem. Isto faz com que <i>qmail-inject</i> adicione a data corrente e o ID do processo no sender da mensagem.

QMAILMFTFILE

Especifica o arquivo que conterá uma lista de e-mails que serão lidos por *qmail-inject*.

Se um ou mais desses e-mails estiverem contidos no campo *To:* ou *Cc:* do cabeçalho do e-mail, *qmail-inject* adicionará o campo de cabeçalho *Mail-Followup-To:* (se já não existir) com todos os endereços de e-mail contidos em *To:* ou *Cc:*.

QMAILPLUSDOMAIN

Seu conteúdo tem precedência sobre o valor definido no arquivo de controle *plusdomain*.

QMAILSHOST/QMAILUSER

Seu conteúdo tem precedência sobre o valor definido para o *sender* da mensagem, quando enviada utilizando *qmail-inject*.

RECIPIENT

Endereço de e-mail do recipiente da mensagem.

RELAYCLIENT

Se setado, torna nulo o conteúdo do arquivo *rcpthosts* e permite ao usuário enviar e-mails para qualquer domínio local ou remoto.

RPLINE

Linha completa de *Return-Path*: utilizada no cabeçalho do e-mail.

SENDER

Endereço de e-mail utilizado como *sender* do e-mail.

UFLINE

Linha completa de *From*: utilizada no cabeçalho do e-mail no formato UUCP

Sistema de Aliases/Forwards

O qmail possui um avançado, versátil e seguro sistema de aliases e forwards, e ainda pode aproveitar-se de mecanismos de aliases/forwards de terceiros, para facilitar migrações.

dot-qmail

`/var/qmail/alias/.qmail-alias`

Aonde *alias* é o nome do alias que será interpretado pelo qmail.

Normalmente é um arquivo contendo um ou mais endereços de e-mail para os quais as mensagens devem ser redirecionadas.

É importante lembrar que, `~/alias/.qmail-alias` só será alcançado caso não exista um usuário local identificado por *alias*. Isto significa que, se existir um usuário *telles* e for criado um alias `~/alias/.qmail-telles`, este alias nunca será alcançado.

Linhas que iniciam com “#” (comentários) são ignoradas.

Ex: criando o alias *rodrigo*.

`# echo telles@domain.net > /var/qmail/alias/.qmail-rodrigo`

Isso fará o qmail redirecionar todas as mensagens enviadas a *rodrigo* para *telles@domain.net*.

É possível utilizar vários e-mails para redirecionamento de mensagens dentro do arquivo de alias, bastando para isso, que cada e-mail esteja em uma única linha:

`# cat /var/qmail/alias/.qmail-galera`

telles@tubes.br

cleri@tdk.ft

langona@ls.up

kerne@games.md

alvsan@lnx.pt

Ainda é possível utilizar scripts ou programas dentro dos arquivos *.qmail-alias*.

Ex:

```
# echo '| /home/telles/lixo.sh' > /var/qmail/alias/.qmail-spam
```

Desta forma, todas as mensagens enviadas para *spam* serão tratadas pelo script */home/telles/lixo.sh*, o que provavelmente irá deletar a mensagem com algo do tipo:

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

```
cat > /dev/null
```

```
exit 99
```

NOTA: Caso deseje criar aliases que possuam “.” (ponto) no meio do nome (Ex: r.p.telles), substitua “.” por “:” (dois pontos).

Ex:

```
$ echo telles@tubes.br > /var/qmail/alias/.qmail-r:p:telles
```

\$HOME/.qmail- alias

Aonde *alias* é o nome do alias criado pelo usuário local.

Desta forma, o usuário pode criar seus próprios aliases, sem depender do administrador de sistemas.

Exemplo 1: *\$HOME* da usuária *aline*.

```
$ echo joazinho > $HOME/.qmail-encontro
```

Quando for enviado um e-mail para *aline-encontro*, qmail redirecionará a mensagem para o usuário *joazinho*.

Ainda é possível redirecionar mensagens para caixas de e-mail no formato *mbox*.

Exemplo 2: *\$HOME* da usuária *aline*.

```
$ echo ./Mailbox.encontros > $HOME/.qmail-encontros
```

Desta forma, todos os e-mails enviados para *aline-encontros* serão armazenados na caixa de e-mail *\$HOME/Mailbox.encontros* no formato *mbox*.

Quando uma mensagem para *telles-amigos-smith* é enviada, *qmail-lspawn* checa se o usuário *telles* existe, se existir *qmail-local* verifica no *\$HOME* do usuário *telles*, se *.qmail-amigos-smith* existe, se não existir ou se estiver vazio, checa por *qmail-amigos-default* e por final *.qmail-default*; *qmail-local* seguirá as instruções contidas no primeiro arquivo encontrado que tenha conteúdo diferente de vazio. Se o usuário *telles* não existir, *qmail-lspawn* passa o controle da entrega para o usuário *alias* antes de rejeitar a mensagem.

Quando *qmail-local* encaminha uma mensagem seguindo instruções contidas em um arquivo *\$HOME/.qmail-alias* (ou *\$HOME/.qmail-default*), sempre verifica a existência de um arquivo do tipo *\$HOME/.qmail-alias-owner*; se o arquivo existir, *qmail-local* utilizará *usuario-alias-owner@dominio* como sendo o *sender* (Return-Path:) da mensagem. Caso contrário, *qmail-local* mantém o endereço de sender original da mensagem.

Excessão: *qmail-local* sempre mantém o endereço original de *sender* se este for vazio ou contiver “#@[]” (mensagens de erro – bounces).

qmail-local também suporta a utilização de *sender variável* (VERP – Variable Envelope Return Paths). Se *\$HOME/.qmail-alias-owner* e *\$HOME/.qmail-alias-owner-default* existirem, *qmail-local* utilizará *usuario-alias-owner-@dominio-@[]* como *sender* de mensagens encaminhadas. Isto fará com que o recipiente da mensagem - *recipiente@host* - veja o *sender* como sendo *usuario-alias-owner-recipiente=host@dominio*.

O arquivo *\$HOME/.qmail* pode ser utilizado para alterar o método padrão de entrega de mensagens para o usuário.

Se *\$HOME/.qmail* estiver completamente vazio ou não existir, *qmail-local* utilizará o método padrão definido pelo administrador do sistema em *qmail-start* (defaultdelivery); normalmente o método padrão é *mbox* e todas as mensagens serão adicionadas ao arquivo *\$HOME/Mailbox* neste padrão.

\$HOME/.qmail pode conter espaços em branco e tabs no final de cada linha; Linhas em branco são permitidas, desde que não sejam a primeira linha do arquivo.

Se *\$HOME/.qmail* for executável (possuir o bit de execução – x – ligado) e contiver qualquer linha de programa, instruções de *Mailbox* ou *Maildir*, *qmail-local* suspende a entrega e devolve a mensagem à fila de e-mails.

Exemplo 3: *\$HOME* do usuário *telles*.

```
$ echo './Maildir' > $HOME/.qmail
```

É importante lembrar que, será necessário criar o diretório *Maidir* e sua estrutura, utilizando o programa *maildirmake*.

Ainda é possível utilizar scripts e/ou programas dentro dos arquivos *dot-qmail*.

Exemplo 4: *\$HOME* do usuário *telles*.

```
$ cat $HOME/.qmail
```

```
# Programa para filtragem de lixo
```

```
|./lixo.sh
```

```
&telles@tubes.sp
```

```
./Maildir/
```

No exemplo acima, todas as mensagens enviadas para o usuário *telles*, serão submetidas ao programa *lixo.sh* e se passarem desta etapa, serão encaminhadas ao e-mail *telles@tubes.sp* e entregues no *Maidir* do usuário.

Se *.qmail* possuir permissões de leitura para todos usuários ou grupos do sistema, ou se o diretório *\$HOME* do usuário possuir a flag *sticky* ligada, *qmail-local* suspende a entrega e devolve a mensagem à fila de e-mails.

A utilização da flag *sticky* (*chmod +t \$HOME*) deve ser utilizada para manutenção e edição do arquivo *.qmail* com segurança, fazendo com que novas mensagens que alcancem a caixa de correio do usuário, voltem para a fila e sejam entregues mais tarde, quando a flag *sticky* não estiver mais setada (*chmod -t \$HOME*) .

Exemplo 5:

```
$ chmod +t $HOME
```

```
$ /var/qmail/bin/qmail-local -n $USER ~ $USER " " " " ./Maildir/
```

Warning: home directory is sticky.

```
maildir ./Maildir/
```

```
did 1+0+0
```

```
$ chmod -t $HOME
```

```
$ /var/qmail/bin/qmail-local -n $USER ~ $USER " " " " ./Maildir/
```

```
maildir ./Maildir/
```

```
did 1+0+0
```

qmail-users - assign/cdb

É possível criar uma tabela de usuários do qmail associada a usuários do sistema.

Este sistema utiliza os arquivos *assign* e *cdb* que estão localizados no diretório */var/qmail/users* e é muito similar aos sistemas de aliases já mencionados, com a diferença de utilizar métodos e programas diferentes para a mesma tarefa.

O arquivo *assign* contém os mapeamentos de usuários qmail para usuários do sistema, um em cada linha; *cdb*, é um arquivo binário gerado pelo programa *qmail-newu* a partir de *assign*.

assign possui o formato:

=endereço:usuário:uid:gid:diretório:traço:alias:

Ainda é possível utilizar o sinal “+” (adição) como curinga no lugar de “=” (igual), que fará o qmail localizar *endereço* no início do endereço de recipiente.

Ex:

```
# cat /var/qmail/users/assign
```

```
=telles:rodrigo:500:500:/home/rodrigo:::
```

```
+news-:rodrigo:500:500:/home/rodrigo:--:
```

```
.
```

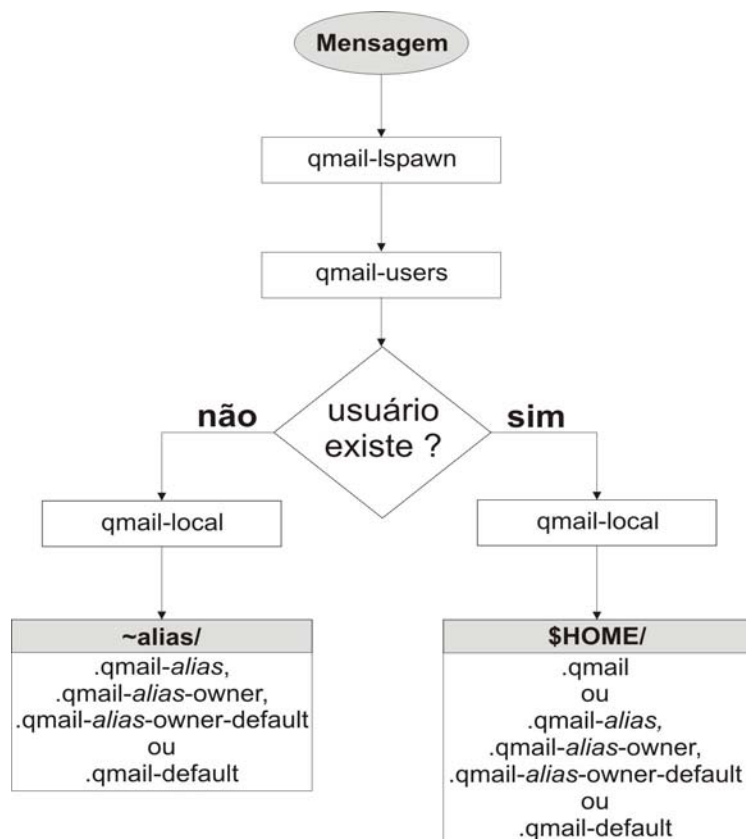
```
# /var/qmail/bin/qmail-newu
```

E-mails enviados para *telles* serão entregues para o usuário *rodrigo* e e-mails enviados para qualquer endereço que comece com “*news-*” também serão entregues ao usuário *rodrigo*.

Desta forma, um e-mail enviado para *news-unix* será tratado por */home/rodrigo/.qmail-unix*.

NOTA: A última linha do arquivo *assign* sempre deve ser um único “.” (ponto).

Entrega local:



O programa dot-forward

O programa dot-forward estende os recursos de encaminhamento do qmail, permitindo-o utilizar os arquivos de forward (.forward) do *sendmail*, tornando migrações transparentes para os usuários.

dot-forward

dot-forward [-nN] .forward

Se *.forward* existir, dot-forward encaminha mensagens de entrada de acordo com instruções no padrão sendmail contidas em *.forward*.

Exemplo 1:

Dentro do arquivo *\$HOME/.qmail* do usuário:

| dot-forward .forward

./Maildir/

Não esqueça de incluir uma linha logo após a instrução dot-forward, informando ao programa *qmail-local* que a mensagem deve ser entregue na caixa de e-mail do usuário se dot-forward não encontrar o arquivo *.forward*, caso contrário o usuário local não receberá novas mensagens.

Exemplo 2:

É possível informar vários arquivos de forward:

| dot-forward .forward .forward1 .forward2**./Maildir/**

dot-forward vai executar as instruções do primeiro arquivo de forward que existir e contiver instruções, caso contrário, testa cada arquivo até o final da lista.

Normalmente, dot-forward sai com código de erro 99, desta forma *qmail-local* ignora as próximas instruções contidas no arquivo *.qmail*.

Se o nome do usuário local estiver listado em *.forward* ou *.forward* não existir, dot-forward sai com código de erro 0 (zero) e *qmail-local* processará o restante das instruções contidas no arquivo *.qmail* normalmente.

Se dot-forward encontrar algum erro ao tentar ler *.forward*, sai com código de erro 111, fazendo com que o e-mail volte para a fila; *qmail-local* tentará entregá-lo novamente mais tarde.

Exemplo 3:

.forward pode conter várias linhas, cada uma contendo vários endereços de e-mail

\$ cat .forward

telles, kerne, langona

Joao Dias <joao@dias.com>

Exemplo 4:

Ainda é possível utilizar comandos e comentários dentro de *.forward*:

\$ cat .forward

Aviso de férias para Telles

telles, "[vacation telles"

dot-forward entregará novos e-mails para *telles* e em seguida os encaminhará ao programa *vacation*, juntamente com as variáveis de ambiente *\$UFLINE*, *\$RPLINE* e *\$DTLINE*, utilizando *telles* como parâmetro do programa.

Para utilizar dot-forward de forma global para todos os usuários, basta modificar o arquivo de inicialização do programa *qmail-start* (geralmente em */var/qmail/rc*) para que fique desta forma:

**exec env - PATH="/var/qmail/bin:\$PATH" **

qmail-start '|dot-forward .forward

./Maildir' splogger qmail

Opção	Descrição
-n	Lê <i>.forward</i> e imprime em <i>STDOUT</i> as instruções a serem executadas sem executá-las. Útil para executar dot-forward via linha de comando e visualizar como <i>.forward</i> será interpretado.
-N	Lê e encaminha a mensagem (Padrão).

Notas de compatibilidade:

- dot-forward não suporta métodos “:include:” ou entregas no formato *mbox*. Utilize o método *dot-qmail*.
- dot-forward ignora arquivos *.forward* vazios.
- Se dot-forward encontrar um erro temporário no processamento de um arquivo *.forward*, sai com código de erro 111, fazendo *qmail-local* processar a mensagem novamente mais tarde. *sendmail* assume incorretamente que o arquivo não existe.
- *.forward* precisa ser legível por dot-forward, que normalmente roda com as permissões do usuário local. *sendmail* normalmente é executado pelo usuário *root* e possui acesso total a todos os arquivos.
- *sendmail* manipula “” (aspas) e “\” (contra-barras), violando as *RFC's* 821 e 822. dot-forward não manipula tais caracteres; endereços do tipo “\joao” são tratados da mesma forma que “joao”.
- dot-forward permite comentários nos arquivos *.forward*; versões do *sendmail* anteriores a V8 não permitem comentários nos arquivos *.forward*.

Programas do pacote fastforward

fastforward é um conjunto de programas que permitem ao qmail, utilizar-se de arquivos de *aliases/forwards* (/etc/aliases) e arquivos de tabela de domínios virtuais criados especificamente para o programa sendmail.

fastforward

fastforward [-nNpPdD] cdb

Encaminha cada mensagem recebida, seguindo instruções contidas em *cdb*, criado previamente por *setforward*.

fastforward é geralmente utilizado dentro do arquivo de alias *~alias/.qmail-default*, mas pode ser utilizado também em outros arquivos padrão *dot-qmail*.

Arquivos do tipo *~alias/.qmail-recipient* tem precedência sobre *~alias/.qmail-default*.

Se não existir instruções de forward para o destinatário do e-mail no arquivo *cdb*, fastforward rejeita a mensagem e a devolve ao seu remetente.

Nota de segurança:

Se *cdb* incluir instruções para uma lista de e-mails controlada por outro usuário, este usuário ganhará algumas permissões de fastforward para executar tarefas com as permissões do usuário original. Isso permite que este usuário force o acesso a arquivos que somente o usuário original teria.

Opção	Descrição
-n	Não faz forward da mensagem, apenas imprime instruções que serão seguidas no momento da entrega.

Opção	Descrição
-N	Encaminha a mensagem conforme instruções contidas em <i>cdb</i> (Padrão).
-P	Se fastforward não localizar instruções para o recipiente da mensagem em <i>cdb</i> , sai com código de erro 0 (zero), fazendo com que as próximas instruções do arquivo <i>.qmail-default</i> sejam executadas normalmente. Se fastforward localizar instruções para o recipiente da mensagem em <i>cdb</i> , encaminha a mensagem e sai com código de erro 99, fazendo com que as próximas instruções do arquivo <i>.qmail-default</i> sejam ignoradas.
-P	Se fastforward localizar instruções para o recipiente da mensagem em <i>cdb</i> , encaminha a mensagem e sai com código de erro 0 (zero), caso contrário, rejeita a mensagem e devolve ao seu remetente (Padrão).
-d	Utiliza <i>\$DEFAULT@\$HOST</i> como endereço de recipiente ou <i>\$EXT@\$HOST</i> se <i>\$DEFAULT</i> não estiver setado.
-D	Utiliza <i>\$RECIPIENT</i> como endereço de recipiente (Padrão).

newaliases

Lê uma tabela de aliases a partir do arquivo */etc/aliases*, e converte em uma base de dados *cdb* (*/etc/aliases.cdb*) para ser utilizada pelo programa *fastforward*; *newaliases* é um clone do programa *newaliases* do *sendmail*.

Por segurança, *newaliases* converte a tabela de aliases para */etc/aliases.tmp* e somente depois move */etc/aliases.tmp* para */etc/aliases.cdb*.

Caso ocorra um problema na gravação do arquivo */etc/aliases.tmp*, *newaliases* não tenta mover */etc/aliases.tmp* para */etc/aliases.cdb*, preservando o conteúdo anterior do arquivo inalterado e sem prejudicar entregas que estão acontecendo no momento.

newaliases sempre cria o arquivo */etc/aliases.cdb* com permissão de leitura para todos os usuários do sistema.

newaliases não tenta proteger */etc/aliases.cdb* contra atualizações simultâneas.

Formato do arquivo /etc/aliases

Aliases e forwards podem conter a forma:

root: sysadm

sysadm: telles, kernel

sysop: Just for fun <just@for.fun>

comentários são ignorados

kernel: kerne

executando programas e scripts

script: “| ./script.sh”

Aonde novas mensagens para o usuário *root* serão encaminhadas para *sysadm* que por sua vez serão encaminhadas para os usuários *telles* e *kernel*, aonde *kernel* é um alias para *kerne* e novas mensagens para o usuário *sysop* serão encaminhadas para o e-mail *just@for.fun*.

Quando newaliases encontrar um endereço de recipiente que não possui um nome de domínio (@dominio.com.br) seguido de um “[” (barra vertical), processa o conteúdo logo após a barra vertical como sendo um programa a ser executado. *fastforward* executará *./script.sh* quando receber uma mensagem para *script*.

Qualquer endereço de recipiente que não possua “@dominio.com.br” utilizará os mecanismos *defaulhost*, *defaultdomain* e *plusdomain*.

Instruções de encaminhamentos podem ser separadas em várias linhas, desde que a próxima linha inicie com um espaço em branco, tab ou esteja vazia.

Ex:

hostmaster:

telles,

nobody

Comentários pode ser separados em várias linhas, desde que a próxima linha inicie com um espaço em branco, tab ou esteja vazia.

Ex:

comentários

em varias

linhas são permitidos.

Notas de compatibilidade:

- newaliases não suporta entrega em arquivo, utilize o mecanismo de entrega *dot-qmail* (formato *mbox*).
- Se existir várias entradas de encaminhamentos para o mesmo alias em */etc/aliases*, *sendmail* solta uma mensagem de erro na tela; *fastforward* utilizará silenciosamente a primeira instrução.
- *sendmail* processa aspas “”(aspas) e “\” (contra-barras) em */etc/aliases*, violando as *RFC's 821* e *822*; newaliases não processa tais caracteres.
- *sendmail* não suporta aliases por host, sendo assim, processa *user@host.com.br* da mesma forma que *user*, se *host.com.br* for local.
- Grupos de endereços não são permitidos.
- Várias versões do *sendmail* fazem coisas estranhas quando encontram aliases circulares; nestes casos, *fastforward* entrega a mensagem apenas uma vez.

- Endereços de e-mail podem legitimamente iniciar com “/” (barra) ou “|” (pipe - barra vertical). newaliases processa qualquer instrução que contenha o símbolo “@” e não esteja entre “” (aspas) como um recipiente de e-mail. Aparentemente, *sendmail* tem vários problemas com estes endereços e com comandos que contenham o símbolo “@” (arroba).
- newaliases não permite a utilização de “|” (barra vertical) antes de aspas duplas.
- *sendmail* requer que a “,” (vírgula) esteja na primeira linha de um encaminhamento separado em várias linhas de instruções.
- *sendmail* não permite linhas em branco no meio de uma continuação de encaminhamentos separados em múltiplas linhas.
- *sendmail* não permite comentários divididos em múltiplas linhas.

Arquivos de Include

Se um endereço de recipiente possuir a forma “:include:usuários”, significa que todos os usuários contidos no arquivo usuários devem receber uma cópia da mensagem. Atualmente, *fastforward* lê o arquivo usuários.bin que deve ser criado utilizando-se o programa *newinclude*.

Perceba que o arquivo usuários é lido por *fastforward* e não pelo programa newaliases, desta forma, o administrador não precisa executar newaliases a cada modificação feita em usuários.

Notas de compatibilidade:

- Se o arquivo de include usuários não puder ser lido ou não existir, *sendmail* ignora esta entrada; *fastforward* somente adia a entrega da mensagem, fazendo-a retornar à fila de e-mails.
- *sendmail* não permite espaços em branco no texto da instrução “:include: nome-após-espço”, newaliases permite.
- versões do sendmail anteriores a V8, não retiram as aspas (“quote”) de arquivos “:include:”.

Alias Owners

Se existir um alias *owner*-lista, aonde lista aponta para vários usuários ou outros aliases, *fastforward* altera o *sender* (Return-Path:) de cada mensagem por *owner*-lista, antes de encaminhar a mensagem para os destinatários contidos em lista.

Ex:

humor: telles, kerne, hugo, cleri

owner-humor: telles

Desta forma, novas mensagens para humor serão entregues para telles, kerne, hugo e cleri, e caso algum destes usuários não exista ou ocorra um erro na entrega da mensagem, este erro retornará para o usuário telles listado como owner (dono) da lista humor.

Nota de compatibilidade:

Quando um alias inclui um endereço de recipiente e outros aliases, aonde o mesmo endereço de recipiente também está listado dentro de um dos aliases, *fastforward* enviará a mesma mensagem 2 (duas) vezes, cada uma com um endereço de *sender* diferente. *sendmail* envia a mensagem apenas uma vez, e o *sender* a ser utilizado vai depender muito das fases da lua.

Ex:

owner-humor: telles

humor: telles, kerne, hugo, cleri

friends: kerne, humor

newinclude

newinclude *arquivo-de-lista*

Lê entradas do tipo “*:include: arquivo-de-lista*” no estilo *sendmail* e converte em um formato binário (*arquivo-de-lista.bin*) para ser utilizado por *fastforward*.

newinclude primeiro escreve no arquivo *arquivo-de-lista.tmp* e então move o arquivo para *arquivo-de-lista.bin*.

Caso ocorra um problema na gravação de *arquivo-de-lista.tmp*, *newinclude* não tenta mover *arquivo-de-lista.tmp* para *arquivo-de-lista.bin*, preservando o conteúdo anterior do arquivo *arquivo-de-lista.bin* (caso exista) inalterado.

newinclude sempre cria *arquivo-de-lista.bin* com permissão de leitura para todos os usuários do sistema.

arquivo-de-lista pode conter várias linhas, cada linha pode conter vários endereços de e-mail separados por “,” (vírgula) e ainda outras instruções do tipo “*:include:arquivo-de-lista*”.

Qualquer linha que iniciar com um “#” (comentário) será ignorada.

Nota de compatibilidade:

- *sendmail* lê *arquivo-de-lista* diretamente, *fastforward* necessita do arquivo em formato binário (*arquivo-de-lista.bin*). A estratégia utilizada por *sendmail* pode causar um disastre se o arquivo estiver sendo escrito no disco ao mesmo tempo que *sendmail* estiver lendo-o; o arquivo pode ser truncado e ainda incluir “sugeiras” no meio dos endereços de e-mail da lista; se o sistema travar neste momento, o arquivo será preenchido com todo tipo de lixo.
- *newinclude* não suporta entrega de mensagens em arquivo ou através de programas em arquivos “*:include:*”. Utilize o mecanismo *dot-qmail*.
- Versões do *sendmail* anteriores a V8 não permitem comentários em arquivos “*:include:*”.

printforward

Lê uma base de dados (cdb) de encaminhamentos via *STDIN* e imprime em *STDOUT* todas as instruções de encaminhamentos em um formato aceito por *setforward*.

Ex:

```
$ /var/qmail/bin/printforward < /etc/aliases.cdb
```

```
mailer-daemon@:
```

```
, &postmaster@myqmail.box
```

```
;
```

```
postmaster@:
```

```
, &root@myqmail.box
```

```
;
```

```
root@:
```

```
, &telles@myqmail.box
```

```
;
```

```
humor@:
```

```
, &cleri@myqmail.box
```

```
, &hugo@myqmail.box
```

```
, &kerne@myqmail.box
```

```
, &telles@myqmail.box
```

```
;
```

```
humor@: ?owner-humor@myqmail.box;
```

```
owner-humor@:
```

```
, &telles@myqmail.box
```

```
;
```

printmaillist

Lê uma lista de e-mails no formato binário via *STDIN* e imprime todas as instruções de encaminhamento em *STDOUT* em um formato aceito por *setmaillist*.

Ex:

```
$ /var/qmail/bin/printmaillist < lista.bin
```

```
&rodrigo@nosp.am
```

```
&root@local.net
```

setforward

setforward *cdb tmp*

Lê uma tabela de instruções de encaminhamentos da entrada padrão (*STDIN*) e converte em uma base de dados de encaminhamentos, que pode ser utilizada por *fastforward*.

Por segurança, *setforward* escreve a base de dados no arquivo *tmp* e só depois move *tmp* para *cdb*.

Caso ocorra um problema na gravação do arquivo *tmp*, *setforward* não tenta mover *tmp* para *cdb*, preservando o conteúdo anterior do arquivo inalterado.

A base de dados *cdb* é portátil entre computadores diferentes, tornando possível a utilização da mesma base por computadores distintos.

As instruções de redirecionamentos devem conter um alvo, seguido de “:” (dois pontos), uma série de comandos e finalizar com “;” (ponto-e-vírgula):

alvo: comando1, comando2, comando3, ...;

Cada comando, pode ser um ou vários endereços de recipiente separados por “,” (vírgula), um endereço de owner, uma lista de endereços externos ou um programa.

Ex:

sysadm@host.com: telles@space.net, sysadm@nosp.am;

Aonde, e-mails para *sysadm@host.com* serão encaminhados para *telles@space.net* e *sysadm@nosp.am*.

setforward ignora linhas que iniciam com “#” (comentários), finais de linha (permitindo dividir instruções de forwards em mais de uma linha), espaços em branco e tabs (pode-se utilizar de contra-barras para escapar espaços em branco, tabs, vírgulas, caracteres especiais, etc).

Quando *fastforward* recebe uma mensagem contendo o destinatário *user@host.dom*, verifica primeiro a existência do alvo *user@host.dom*, depois *@host.dom* e por fim *user@*. Obedecerá as regras de encaminhamento do primeiro alvo que encontrar.

Todos os comandos para um alvo devem ser listados em uma única instrução; com exceção de endereços de owner.

Se um comando iniciar com um “&”, *setforward* processa o resto do comando como sendo um endereço de recipiente.

Ex:

root@nosp.am: &ni@nosp.am;

setforward encaminhará cada nova mensagem com destino a *root@nosp.am* para *ni@nosp.am*. O endereço de recipiente deve ter a forma *user@dominio.com* e não pode ultrapassar 800 bytes (800 caracteres).

Endereços de owner

Se um comando iniciar com o sinal “?” (interrogação), *setforward* processa o resto do comando como sendo um endereço de owner:

suporte@nosp.am: ?owner-suporte@nosp.am;

fastforward utilizará o endereço *owner-suporte@nosp.am* como *sender* (Return-Path:) de todos os e-mails encaminhados a partir de *suporte@nosp.am*.

Normalmente, se uma mensagem é encaminhada para um recipiente com problemas, a mensagem volta com erro pa-

ra o endereço utilizado como *sender*.

Listas externas de e-mails

Se um comando iniciar com um “.” (ponto) ou com uma “/” (barra), *fastforward* processa o resto do comando como sendo o nome de um arquivo binário contendo uma lista de e-mails gerada pelo programa *setmaillist*.

Ex:

suporte@nosp.am: /etc/lists/suporte.bin;

fastforward irá ler o conteúdo do arquivo **/etc/lists/suporte.bin** e seguir as instruções de encaminhamento contidas neste arquivo. O arquivo precisa ser acessível por qualquer usuário e por *fastforward*.

Programas

Se um comando iniciar com um “|” (pipe – barra vertical) ou um “!” (sinal de exclamação), *fastforward* processa o resto do comando como sendo o nome de um programa a ser executado.

Ex:

root-cmd@: | /bin/monitor;

Se utilizado “|”, *fastforward* encaminhará a mensagem para o programa *monitor* quando receber uma mensagem para *root-cmd*.

Se utilizado “!”, *fastforward* age da mesma forma, exceto que enviará também as variáveis de ambiente *\$UFLINE*, *\$RPLINE* e *\$DTLINE* ao programa *monitor*.

Duplicidades

Quando *setforward* está montando a lista de recipientes para cada e-mail, ele segue a lista de recipientes que será utilizada para os encaminhamentos; se detectar duplicidades de instruções de encaminhamento, ignora a segunda ocorrência da instrução e continua o processamento.

Ex:

everybody@nosp.am: programmers@nosp.am, testers@nosp.am;

programmers@nosp.am: root@nosp.am, devel@nosp.am;

testers@nosp.am: root@nosp.am, fred@nosp.am;

Mensagens para *everybody@nosp.am* serão encaminhadas para *root@nosp.am* apenas uma vez. *setforward* se encarrega de evitar loops internos.

Exceção: Se o alvo possuir um endereço de owner, comandos para este alvo serão considerados de forma diferente.

Ex:

everybody@nosp.am: programmers@nosp.am, testers@nosp.am;

programmers@nosp.am: root@nosp.am, devel@nosp.am;

testers@nosp.am: root@nosp.am, fred@nosp.am;

testers-owner@nosp.am: root@nosp.am

Mensagens para *everybody@nosp.am* serão encaminhadas para *root@nosp.am* 2 (duas) vezes; uma vez através de *programmers@nosp.am* e outra vez através de *testers@nosp.am*. Isto ocorre por causa da existência de *testers-ow-*

ner@nosp.am, que torna *root@nosp.am* especial. *setforward* elimina endereços duplicados para aliases regulares, mas não para listas de e-mails, indicada acima pela existência de *testers-owner*.

setmaillist

setmaillist *arquivo.bin* *arquivo.tmp*

Lê uma lista de e-mails via *STDIN*, escreve em formato binário em *arquivo.tmp* e move *arquivo.tmp* para *arquivo.bin*.

Caso ocorra um problema na gravação de *arquivo.tmp*, *setmaillist* imprime uma mensagem de erro em *STDOUT* e não tenta mover *arquivo.tmp* para *arquivo.bin*, preservando o conteúdo anterior do arquivo *arquivo.bin* (caso exista) inalterado.

arquivo.bin e *arquivo.tmp* devem obrigatoriamente estar localizados no mesmo sistema de arquivos.

O arquivo binário gerado por *setmaillist* é portátil entre computadores diferentes, tornando possível a utilização do mesmo arquivo por computadores distintos.

setmaillist sempre cria *arquivo.bin* com permissão de leitura para todos os usuários do sistema.

Formato de entrada

Se uma linha iniciar com um “.” (ponto) ou com uma “/” (barra), *setmaillist* processa o resto da linha como sendo um arquivo de include.

Se uma linha iniciar com um “&”, letras ou números, *setmaillist* processa o resto da linha como sendo um endereço de recipiente.

Cada endereço de recipiente deve ter a forma *user@dominio.com* e não pode ultrapassar 800 bytes (800 caracteres).

setmaillist ignora linhas em branco, linhas que iniciam com um “#” (comentário), espaços e tabs no final de linhas.

Ex:

admin@nosp.am

&abuse@localhost.net

/etc/mail/amigos.bin

postmaster@local.net

O programa checkpassword

O programa *checkpassword* é utilizado para efetuar a autenticação de usuários utilizando o arquivo */etc/password*.

checkpassword

checkpassword *programa*

Geralmente utilizado em conjunto com o programa *qmail-popup*, *checkpassword* recebe via file descriptor 3 o nome de usuário e senha que devem ser verificados no arquivo */etc/password*.

Se a senha for aceita para o usuário, *checkpassword* executa *programa*. *programa* consiste em um ou mais argumentos.

Se a senha não for aceita, retorna código de erro 1.

Se ocorrer um problema temporário na checagem da senha, retorna código de erro 111.

Retorna código de erro 2 (dois) para qualquer outra situação inesperada.

Exemplo 1:

```
# /var/qmail/bin/qmail-popup nosp.am /bin/checkpassword pwd
```

```
+OK <1484.1083201875@nosp.am>
```

```
user ana
```

```
+OK
```

```
pass senha
```

```
/home/ana
```

Se usuário e senha estiverem corretos, checkpassword executa o comando *pwd* utilizando o usuário *ana*.

Exemplo 2:

```
# /var/qmail/bin/qmail-popup nosp.am /bin/checkpassword qualquer-comando
```

```
+OK <1504.1083202316@nosp.am>
```

```
user zica
```

```
+OK
```

```
pass brava
```

```
-ERR authorization failed
```

Se usuário e senha estiverem corretos, mas *qualquer-comando* não for um comando válido ou não puder ser localizado, checkpassword falha na autenticação.

Exemplo 3:

```
# /var/qmail/bin/qmail-popup nosp.am /bin/checkpassword pwd
```

```
+OK <1511.1083202756@nosp.am>
```

```
user paula
```

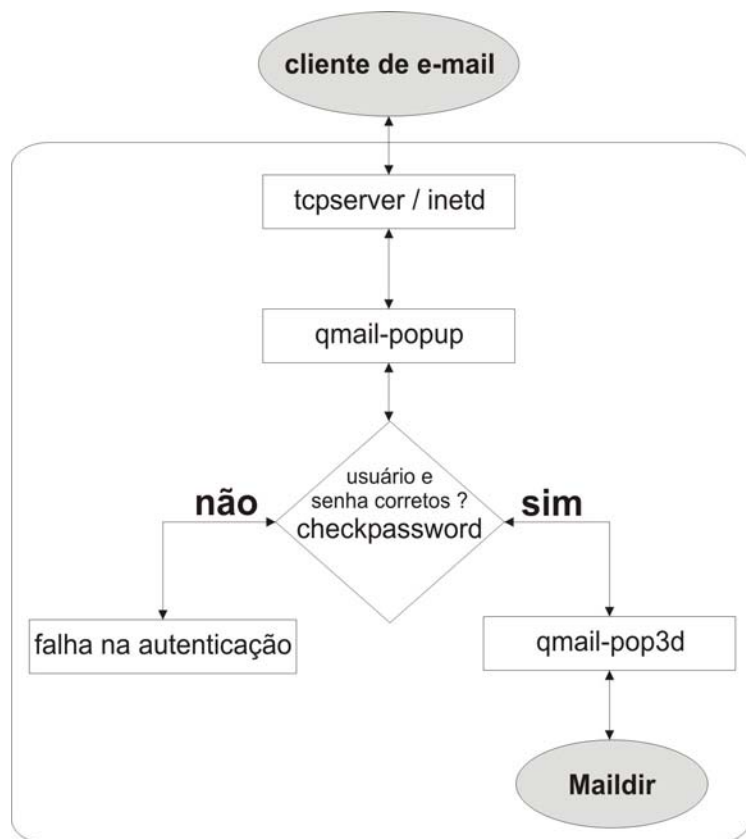
```
+OK
```

```
pass lalala
```

```
-ERR authorization failed
```

Se usuário e senha não estiverem corretos, checkpassword falha na autenticação e não executa o comando *pwd*.

Sessão de autenticação POP3:



Sistema de entrega

Mailbox

Método de entrega de mensagens mais comumente utilizado, que consiste em um arquivo no formato *mbox* (Ex: Mailbox) aonde são armazenadas todas as mensagens, tanto mensagens lidas quanto não lidas; o status da mensagem pode ser identificado por *FLAGS* setadas no campo "*Status:*" no cabeçalho de cada mensagem.

Uma mensagem formatada no padrão *mbox*, inicia com uma linha utilizando a notação "*From* " (*From* seguido de um espaço em branco) e finaliza com uma linha em branco. O conteúdo entre "*From* " e a linha em branco deve ser uma mensagem formatada conforme a *RFC 822*.

Ex:

From root@nosp.am Sun Jul 28 11:05:34 2004

Return-Path: <root@nosp.am>

Delivered-To: root@localnet.org

Date: 28 Jul 2004 11:05:34 -0300

From: Imortal <root@nosp.am>

To: root@localnet.org

Olá, esta é uma mensagem que vai para o guia de consulta sobre qmail.

Meus cumprimentos.

A última linha da mensagem deve ser uma linha em branco, sem espaços em branco ou tabs.

Como a mensagem é entregue

Primeiramente é criada a linha “*From* “ contendo o *sender* da mensagem e a data corrente. Se o *sender* for vazio (Ex: se for uma mensagem do tipo bounce), *qmail-local* utiliza *MAILER-DAEMON* como *sender*. Se o *sender* possuir espaços em branco, tabs ou newlines, *qmail-local* os substitui por hífen.

A mensagem é copiada para o arquivo, e todas as linhas são filtradas utilizando a citação “>*From*”, que significa que cada linha não deve ser identificada como uma linha de “*From* “. Se uma linha iniciar com “*From*”, será inserido o caracter “>” no início da linha. Se iniciar com “>*From*”, será transformada em “>>*From*” e assim por diante.

Finalmente, é inserida uma linha em branco no final da mensagem. Se a última linha da mensagem for uma linha parcial, serão adicionadas 2 (duas) linhas em branco, caso contrário somente uma linha será adicionada.

Como a mensagem é lida

O leitor de mensagens deve procurar primeiramente pela linha que inicia com “*From* “. Qualquer linha que inicie com “*From* “ indica o início de uma mensagem. O leitor de mensagens não deve se esforçar em tomar vantagem do fato de cada linha de “*From* “ ser precedida por uma linha em branco.

Uma vez que o leitor de mensagens encontra o início de uma mensagem, extrai o *sender* e a data de entrega da linha de “*From* “, então lê a mensagem até a próxima linha de “*From* “ ou até o final do arquivo. Extrai a última linha em branco e deleta os caracteres “>” que possivelmente foram inseridos no início das linhas que iniciam com “*From* “ ou “>*From* ” e assim por diante. O resultado é uma mensagem formatada conforme a RFC 822.

Existem muitas variantes do formato *mbox*, a variante utilizada neste documento é conhecida como *mboxrd* e foi popularizada por *Rahul Dhesi* em Junho de 1995.

Detalhes de travamento de arquivo

Existem muitos mecanismos de travamento de arquivos no formato *mbox*. *qmail-local* sempre utiliza *flock* em sistemas que suportam este mecanismo; caso contrário, utiliza *lockf*.

A data de entrega na linha de “*From* ” não especifica um fuso horário. *qmail-local* sempre cria a data de entrega no formato *GMT* (Greenwich Mean Time - Horário Médio de Greenwich), desta forma, arquivos *mbox* podem ser transportados de um fuso horário para outro sem necessidade de alterações da data.

Se *mtime* (hora da última modificação) de um arquivo *mbox* não vazio for maior que *atime* (hora do último acesso), significa que o arquivo *mbox* tem novo e-mail. Se *mtime* for menor que *atime*, significa que o novo e-mail já foi lido. Se *mtime* for igual a *atime*, não há forma de determinar se existem novos e-mails ou não no arquivo *mbox*; para estes casos, a solução é o leitor de e-mails setar artificialmente *mtime* com o valor de *atime* + 1. Desta forma, o arquivo *mbox* somente terá novas mensagens se *atime* for menor ou igual a *mtime*.

Alguns leitores de e-mails adicionam o campo “*Status:*” em cada mensagem, para indicar que a mensagem já foi lida.

Pelo fato de todas as mensagens serem armazenadas em um único arquivo, o método *mbox* torna-se bastante vulnerável a falhas de corrupção, sejam por métodos de acesso/escrita simultâneos ou por travamentos do sistema no momento de entrega de novas mensagens.

Por padrão, *qmail-local* adiciona novas mensagens no final do arquivo *Mailbox*.

O formato *mbox* não pode ser utilizado com segurança em sistemas de arquivo *NFS* (Network File System).

Maildir

Método de entrega de mensagens que consiste em um diretório (Maildir) com subdiretórios (*new*, *cur* e *tmp* - devem estar localizados no mesmo sistema de arquivos), aonde cada mensagem é armazenada em um arquivo independente.

Maildir é mais eficiente que o tradicional formato *mbox*, pois não causa truncamentos de arquivos – seja por acesso simultâneo ou travamento do sistema no momento da entrega - e problemas de travamento de caixa postal.

Desta forma, um computador pode ter 2 (dois) ou mais programas entregando mensagens simultaneamente para o mesmo usuário sem nenhum problema.

O diretório *tmp* é utilizado no momento da entrega como destino temporário de novas mensagens, e somente depois que o sistema de entrega consegue escrever a mensagem com sucesso dentro de *tmp*, a mensagem é copiada para dentro do diretório *new* e então removida do diretório *tmp*. Cada arquivo dentro do diretório *new* é uma nova mensagem. O *mtime* do arquivo será a data de entrega da mensagem. A mensagem é entregue sem nenhuma linha adicional de “*From*”, sem nenhuma citação do tipo “>*From*” e sem nenhuma linha em branco adicional no final do arquivo. A mensagem normalmente está formatada conforme a *RFC 822*, iniciando com uma linha de “*Return-Path*” e uma linha “*Delivered-To*”, mas também pode conter dados binários arbitrários. A mensagem não precisa necessariamente finalizar com uma linha em branco.

As mensagens são armazenadas em arquivos com nome no formato *timestamp.pid.nome-do-host*.

Ex:

1082564410.2518.mybox.nosp.am

Aonde, *1082564410* é o número de segundos contados a partir de 00:00:00 01-01-1970 UTC, *2518* é o número do processo que gerou a entrega e *mybox.nosp.am* é o nome local da máquina.

O formato Maildir pode ser utilizado sem problemas em sistemas de arquivo *NFS* (Network File System).

Funcionamento do sistema de entrega

Mensagens entregues por *qmail-local* são feitas em 6 etapas:

1. Altera o diretório corrente para o diretório Maildir do usuário (*\$HOME/Maildir*).
2. Utiliza a função *stat()* do UNIX para determinar se o arquivo *tmp/timestamp.pid.nome-do-host* existe.
3. Se o arquivo existir, aguarda 2 (dois) segundos e tenta novamente utilizando o *timestamp* atualizado, se falhar, tenta mais uma vez, e em caso de falha, abandona a entrega e retorna um erro.
4. Se o arquivo não existir, cria o arquivo *tmp/timestamp.pid.nome-do-host*.
5. Armazena a mensagem recebida no arquivo *tmp/timestamp.pid.nome-do-host* criado anteriormente.
6. O arquivo é copiado com o mesmo nome para dentro do diretório *new* e só então é removido do diretório *tmp*.

Mensagens já lidas pelo cliente de e-mail (MUA – Mail User Agent), são movidas do diretório *new* para dentro do diretório *cur*. Tal ação só é executada caso o cliente de e-mail esteja configurado para deixar cópia de mensagens no servidor. Normalmente, a mensagem é removida após ser lida.

O leitor da mensagem também deve verificar por mensagens antigas dentro do diretório *tmp* e remove-las. O arquivo

é considerado antigo e pode ser removido com segurança se não foi acessado a 36 horas atrás.

É uma boa idéia para leitores de mensagens, ignorarem arquivos dentro de *new* e *tmp* que iniciem com “.” (ponto).

Leitores de e-mail que suportam nativamente o formato *Maldir*, utilizam a variável de ambiente *\$MAILDIR* para localizar corretamente o diretório *Maildir* do usuário.

Quando a mensagem é movida para o diretório *cur*, o nome do arquivo é alterado para:

timestamp.pid.nome-do-host:info

O *timestamp*, o *pid* e *nome-do-host* continuam os mesmos.

:info é equivalente ao campo “*Status:*” utilizado no cabeçalho de mensagens no formato *mbx*.

info consiste em 2 (dois) campos separados por “,” (vírgula), o primeiro é um número e o segundo é uma *FLAG*, utilizada para ajudar a identificar o status da mensagem aos clientes de e-mail (MUA).

Ex:

1082564410.2518.mybox.nosp.am:2,S

Se o campo numérico contiver o número 1 (hum), significa que a *FLAG* logo após a vírgula é do tipo experimental.

Se o campo numérico contiver o número 2 (dois), significa que a *FLAG* logo após a vírgula é padrão, do tipo:

Flag	Descrição
P (passed)	O usuário encaminhou/rebateu esta mensagem para outro recipiente.
R (replied)	O usuário respondeu a esta mensagem.
S (seen)	O usuário leu esta mensagem.
T (trashed)	O usuário moveu esta mensagem para a lixeira (trash) e será removida na finalização da sessão.
D (draft)	O usuário considera esta mensagem um rascunho/esboço (draft).
F (flagged)	Flag definida pelo usuário.

Programas do pacote ucspi-tcp

O pacote *ucspi-tcp* (*Unix Client-Server Program Interface*) é um conjunto de programas específicos para manipulação de conexões de rede utilizando o protocolo TCP.

addcr

Insere um *CR* (carriage return - \015) antes de cada *LF* (line feed - \012) no final de cada linha de entrada.

Não insere o *CR* em linhas que possuem um final de linha parcial.

Vários programas *unix2dos* e *dos2unix* tem problemas com linhas longas e caracteres nulos, *addcr* não tem problemas com tais caracteres.

argv0

argv0 *programa* *argumento-zero* [*argumentos*]

Executa *programa* e seus argumentos utilizando *argumento-zero* como se fosse o nome de *programa*.

Alguns programas utilizam o argumento zero (argv[0] – nome do programa) como argumento especial e argv0 permite setar *argumento-zero* para argv[0].

Ex :

\$./programa.bin

Meu nome é: ./programa.bin

\$ argv0 ./programa.bin meu-novo-nome

Meu nome é: meu-novo-nome

date@

date@ *host-remoto*

Imprime a data de *host-remoto*.

Conecta em *host-remoto* na porta 13/TCP (daytime) e imprime a data corrente do sistema.

Remove o CR (carriage return - \015) da entrada recebida e converte caracteres que não podem ser impressos na tela em um formato visível.

Se *host-remoto* não for informado, date@ irá conectar em *localhost*.

Ex:

tcpserver -RH10 0 13 /bin/date

\$ date@

Wed Aug 4 23:17:15 BRT 2004

delcr

Remove o CR (carriage return - \015) do final de cada linha de entrada, se o CR estiver presente.

Não remove o CR em linhas que possuem um final de linha parcial.

Vários programas *unix2dos* e *dos2unix* tem problemas com linhas longas e caracteres nulos, delcr não tem problemas com tais caracteres.

finger@

finger@ *host-remoto* *usuário*

Busca informações de *usuário* em *host-remoto*.

Conecta em *host-remoto* na porta 79/TCP (finger) e tenta obter informações de *usuário*.

Remove o CR (carriage return - \015) da entrada recebida e converte caracteres que não podem ser impressos na tela

em um formato visível.

Se *usuário* não for especificado, *finger@* envia uma linha em branco para *host-remoto*. Alguns computadores respondem com informações de todos os usuários conectados no sistema.

Se *host-remoto* não for informado, *finger@* irá conectar em *localhost*.

fixcrio

fixcrio *programa*

Executa *programa* e insere um retorno de carro (CR – carriage return) no final de cada linha de entrada ou saída de *programa* que não contenha um retorno de carro.

fixcrio é executado como um processo filho de *programa*. Quando *programa* é finalizado, fixcrio também é finalizado.

http@

http@ *host-remoto página-web 80*

Busca *página-web* em *host-remoto* na porta *80* TCP.

http@ envia os comandos

GET */página-web* HTTP/1.0

Host: *host-remoto*

para *host-remoto* na porta TCP especificada e imprime o conteúdo da resposta, removendo o *CR* de cada linha recebida.

Ex:

\$ http@ www.nosp.am /index.php 80

http@ exibirá o conteúdo HTML de *index.php* (caso a página exista) na porta *80* do host *www.nosp.am*.

Se a porta TCP não for especificada, http@ utilizará a porta 80/TCP.

Se *página-web* não for especificada, http@ utilizará “/” (barra).

Se *host-remoto* não for informado, http@ irá conectar em *localhost*.

mconnect

mconnect *host-remoto porta*

Conecta em um servidor SMTP remoto.

Ex:

mconnect 172.17.16.11 25

220 Welcome [Seja bem vindo] ESMTP

mconnect adiciona um *CR* no final de cada linha enviada para *host-remoto*, e imprime em tela qualquer resposta recebida.

Se *porta* não for especificada, mconnect utilizará a porta 25/TCP (SMTP).

Se *host-remoto* não for informado, mconnect irá conectar em *localhost*.

rbldsmtpd

rbldsmtpd [**-BbCc**] [**-R**] [**-t n**] [**-r base -r base1 -r baseN**] [**-a a-base**] *smtpd* [*argumentos*]

Rejeita conexões de IPs que estejam cadastrados em listas negras (RBLs).

Pode ser utilizado em qualquer servidor SMTP que suporte ser executado sob *tcpserver*.

Se a variável de ambiente *\$RBLSMTPD* estiver setada e seu conteúdo não for vazio, *rbldsmtpd* rejeita a conexão e envia o conteúdo dessa variável como mensagem de erro para o cliente; se o conteúdo da variável for vazio, *rbldsmtpd* não rejeita a conexão.

Normalmente, *rbldsmtpd* é executado sob *tcpserver*; você pode utilizar o programa *tcprules* para setar a variável de ambiente *\$RBLSMTPD* para determinados clientes. Se o conteúdo de *\$RBLSMTPD* iniciar com um “-” (hífen), *rbldsmtpd* remove o hífen e utiliza o código de erro 553, podendo assim, rejeitar conexões de clientes legítimos.

Se a variável de ambiente *\$RBLSMTPD* não existir, *rbldsmtpd* procura por *\$TCPREMOTEIP* na base *RBL* e rejeita a conexão se *\$TCPREMOTEIP* estiver listado. *tcpserver* seta a variável *\$TCPREMOTEIP* com o IP do host remoto.

Opções	Descrição
-B	Utiliza o código de erro 451 para IPs cadastrados em listas negras (Padrão).
-b	Utiliza o código de erro 553 para IPs cadastrados em listas negras.
-C	Caso a procura pela lista negra falhe, assume que o IP não está listado e aceita a conexão (Padrão).
-c	Caso a procura pela lista negra falhe, assume que o IP está listado e rejeita a conexão (utiliza código de erro 451, mesmo que -c tenha sido especificado).
-r base	Utiliza <i>base</i> como fonte de consulta para IPs que devem ser bloqueados. <i>base</i> deve ser um endereço IP ou um nome de host. Pode-se especificar várias bases repetindo-se o parâmetro “-r”. Ex: -r <i>bl.spamcop.net</i> -r <i>relays.ordb.org</i> -r <i>spam.halls.net</i> . O endereço IP <i>a.b.c.d</i> estará listado na <i>base</i> , se <i>d.c.b.a.base</i> possuir um registro <i>DNS</i> do tipo <i>TXT</i> . <i>rbldsmtpd</i> utiliza o conteúdo do registro <i>TXT</i> como mensagem de erro para o cliente. Se não for especificada nenhuma base, <i>rbldsmtpd</i> utilizará por padrão a entrada <i>rbl.maps.vix.com</i> . Isso pode ser alterado em próximas versões.
-a a-base	Utiliza <i>a-base</i> como fonte de consulta para IPs que estão listados em <i>base</i> mas mesmo assim devem ser liberados. Pode-se especificar várias bases repetindo-se o parâmetro “-a”. Ex: -a <i>anti-bl.mynet.com</i> -a <i>anti-bl.nosp.am</i> . O endereço IP <i>a.b.c.d</i> será liberado se <i>d.c.b.a.a-base</i> possuir um registro <i>DNS</i> do tipo <i>A</i> . Neste caso, <i>rbldsmtpd</i> não rejeitará a conexão.
-t n	Tempo máximo em segundos que aguarda antes de terminar a conexão com a <i>base</i> de consulta remota (timeout – Padrão: 60 segundos).
smtpd	Software de SMTP a ser utilizado em conjunto com <i>rbldsmtpd</i> . <i>rbldsmtpd</i> somente invoca <i>smtpd</i> caso a conexão seja aceita, e passa o controle da conexão para <i>smtpd</i> .
argumentos	Argumentos a serem utilizados em <i>smtpd</i> .

Se você quiser utilizar a sua própria *RBL* ou anti-*RBL*, utilize o programa *rblDNS* do pacote *djbdns*.

recordio

recordio *programa*

Executa *programa* e imprime a entrada e saída no file descriptor 2.

recordio insere na saída, o *ID* do processo seguido de um “<” para entrada ou “>” para saída. Insere “+” no final de cada linha, um espaço ou um *[EOF]*. O espaço indica que existe um *new line* (\n) na entrada ou na saída, e *[EOF]* indica que chegou ao final da entrada ou da saída.

recordio imprime qualquer pacote de entrada ou saída imediatamente e não tenta combinar pacotes em linhas coerentes.

Ex:

```
$ recordio sed 's/bom/o melhor/' < qmail.txt > /dev/null
```

```
1982 < qmail é bom !
```

```
1982 < [EOF]
```

```
1982 > qmail é o melhor !
```

```
1982 > [EOF]
```

recordio utiliza várias linhas para pacotes longos para garantir que cada linha seja impressa atômicamente.

recordio é executado como um processo filho de *programa*. Quando *programa* é finalizado, **recordio** também é finalizado.

tcpcat

tcpcat *host-remoto porta*

Imprime dados recebidos pela porta TCP *porta* de *host-remoto*.

Ex:

```
$ tcpcat ftp.telles.net 21
```

```
220 FTP Server ready
```

tcpcat pode ser utilizado para transferência de dados binários, pois não aplica nenhum tipo conversão de dados.

tcpclient

```
tcpclient [ -hHrRdDqQv ] [ -i iplocal ] [ -p portlocal ] [ -T x+y ] [ -l nomelocal ] [ -t n ] host porta programa
```

Efetua conexões TCP em hosts remotos.

tcpclient tenta conectar a uma porta TCP em host remoto; se obtiver sucesso, executa *programa* recebendo dados de rede via descriptor 6 e enviando via descriptor 7. Por padrão, cria uma série de variáveis TCP.

Opções	Descrição
-q	Não imprime mensagens de erro na tela.
-Q	Imprime mensagens de erro na tela (Padrão).
-v	Imprime mensagens de erro e de status na tela.
-T $x+y$	Abandona a conexão após $x+y$ segundos. (Padrão: 2+58 segundos). Quando o host possui vários endereços IP, tcpclient tenta conectar no primeiro endereço IP, aguarda x segundos, tenta conectar no segundo endereço IP, aguarda x segundos e assim por diante; logo após a primeira sequência de tentativas, tenta conectar uma segunda vez em cada um dos hosts que falharam, e aguarda y segundos em cada tentativa. É possível omitir $+y$, para tcpclient não executar a segunda tentativa de conexão. Versões de tcpclient anteriores a 0.88, utilizam apenas x (Padrão: 60 segundos).
-i <i>iplocal</i>	Especifica o endereço IP local a ser utilizado como origem da conexão. Aborta com erro se <i>iplocal</i> não estiver disponível. Normalmente tcpclient deixa o sistema operacional escolher o IP.
-p <i>portlocal</i>	Especifica a porta TCP local a ser utilizada como origem da conexão. Aborta com erro se <i>portlocal</i> não estiver disponível. Normalmente tcpclient deixa o sistema operacional escolher a porta.
-l <i>nomelocal</i>	Não tenta resolver <i>nomelocal</i> em DNS. Utiliza <i>nomelocal</i> para a variável de ambiente <i>\$TCPLOCALHOST</i> . Recomendado utilizar 0 (zero) como <i>nomelocal</i> .
-d	Atrasa o envio de dados em uma fração de segundos caso o host remoto responda lentamente (Padrão).
-D	Nunca atrasa o envio de dados. Habilita a opção <i>TCP_NODELAY</i> .
-h	Tenta obter o nome do host remoto via DNS para setar a variável de ambiente <i>\$TCPREMOTEHOST</i> (Padrão).
-H	Não tenta obter o nome do host remoto via DNS e remove a variável de ambiente <i>\$TCPREMOTEHOST</i> caso esteja setada.
-r	Tenta obter informações do host remoto para setar a variável de ambiente <i>\$TCPREMOTEINFO</i> (Padrão).
-R	Não tenta obter informações do host remoto.
-t n	Desiste de obter informações do host remoto para setar <i>\$TCPREMOTEINFO</i> após n segundos (timeout – Padrão: 26 segundos).
host	Alvo remoto a ser utilizado para a conexão. <i>host</i> pode ser 0 (zero), referindo-se ao host local, um endereço IP na forma <i>a.b.c.d</i> ou um nome de host. Se o alvo possuir vários endereços IP, tcpclient tentará conectar em cada endereço IP até obter sucesso na conexão.
porta	Porta TCP alvo do host remoto a ser utilizada. <i>porta</i> deve ser um nome listado em <i>/etc/services</i> ou um número.
programa	Programa a ser executado em caso de sucesso da conexão TCP.

tcprules

tcprules *cdb tmp* < *arquivo-de-regras*

Compila regras que podem ser utilizadas por *tcpserver* para aceitar ou rejeitar conexões TCP em um arquivo *cdb*.

tcprules pode ser utilizado enquanto *tcpserver* estiver rodando. Isto garante que o arquivo *cdb* será atualizado atomicamente.

tcprules compila as regras primeiro em *tmp* e somente depois move *tmp* para *cdb*. Se *tmp* existir, será destruído. *tcprules* precisa ter permissão de escrita nos diretórios que contém *tmp* e *cdb*; *tmp* e *cdb* precisam estar localizados no mesmo sistema de arquivos.

Caso ocorra um problema na gravação do arquivo *tmp*, *tcprules* solta uma mensagem de erro e não tenta mover *tmp* para *cdb*, preservando o conteúdo anterior do arquivo inalterado.

O binário *cdb* é portátil entre computadores diferentes.

Formato das regras

Cada regra deve estar em uma única linha, e linhas que iniciam com “#” (comentários) são ignoradas.

Cada regra deve possuir um *endereço*, seguido de “:” (dois pontos) e uma *lista de instruções* sem espaços extra. Quando *tcpserver* receber uma conexão a partir de *endereço*, seguirá as *instruções* compiladas no arquivo *cdb*.

Endereços

tcpserver procura por regras com vários endereços:

1. *\$TCPREMOTEINFO*@*\$TCPREMOTEIP*, se *\$TCPREMOTEINFO* estiver setado;
2. *\$TCPREMOTEINFO*@=*\$TCPREMOTEHOST*, se *\$TCPREMOTEINFO* e *\$TCPREMOTEHOST* estiverem setados;
3. *\$TCPREMOTEIP*;
4. =*\$TCPREMOTEHOST*, se *\$TCPREMOTEHOST* estiver setado;
5. pequenos prefixos de *\$TCPREMOTEIP* finalizados com um “.” (ponto);
6. pequenos sufixos de *\$TCPREMOTEHOST* iniciados com um “.” (ponto) precedido por “=” (igualdade), se *\$TCPREMOTEHOST* estiver setado;
7. =, se *\$TCPREMOTEHOST* estiver setado;
8. uma string vazia;

tcpserver utilizará a primeira regra que encontrar. É possível utilizar a opção “-p” se desejar que *tcpserver* também procure pela variável de ambiente *\$TCPREMOTEHOST*.

Exemplos de regras:

joao@127.0.0.1:A

18.23.0.32:B

:C

127.:D

Se *\$TCPREMOTEIP* for *10.192.243.70*, *tcpserver* irá seguir as instruções de *C*.

Se *\$TCPREMOTEIP* for *18.23.0.32*, *tcpserver* irá seguir as instruções de *B*.

Se *\$TCPREMOTEIP* for *127.0.0.1* e *\$TCPREMOTEINFO* for *telles*, *tcpserver* irá seguir as instruções de *D*.

Se *\$TCPREMOTEIP* for *127.0.0.1* e *\$TCPREMOTEINFO* for *joao*, *tcpserver* irá seguir as instruções de *A*.

É possível utilizar *tcprulescheck* para checar como *tcpserver* irá interpretar as regras compiladas em *cdb*.

Intervalos de endereços

tcprules processa *1.2.3.37-53:ins* como uma abreviação de regras do tipo *1.2.3.37:ins*, *1.2.3.38:ins*, *1.2.3.39:ins* e assim por diante até *1.2.3.53:ins*. Similarmente, *10.2-3.:ins* é uma abreviação de *10.2.:ins* e *10.3.:ins*.

Instruções

As instruções de uma regra devem iniciar com *allow* ou *deny*. *deny* informa a *tcpserver* para rejeitar a conexão sem executar nenhum programa; *allow* funciona de forma totalmente oposta.

Ex:

telles@127.0.0.1:allow

10.172.9.70:allow,RELAYCLIENT="@domain.org"

:deny

Se a variável de ambiente *\$TCPREMOTEINFO* for *telles* e *\$TCPREMOTEIP* for *127.0.0.1*, *tcpserver* aceitará a conexão.

Se a variável de ambiente *\$TCPREMOTEIP* for *10.172.9.70*, *tcpserver* aceitará a conexão e setará a variável de ambiente *\$RELAYCLIENT* com o conteúdo "*@domain.org*".

Se nenhuma das regras anteriores tomou uma ação (*allow* ou *deny*) com a conexão, a última regra irá negar a conexão.

As "" (aspas) podem ser substituídas por qualquer caractere repetido.

Ex:

10.172.9.70:allow,RELAYCLIENT=/@domain.org/

Qualquer número de variáveis pode ser utilizado, desde que estejam separadas por ",", (vírgula).

Ex:

10.172.9.70:allow,RELAYCLIENT="@domain.org",RBLSMTDP="",TCPLOCALHOST=/magneto.nosp.am
/

tcprulescheck

tcprulescheck cdb

Checa qual ação será tomada por *tcpserver*, baseado no conteúdo de *cdb* e nas variáveis de ambiente setadas.

Imaginando que o conteúdo do arquivo *cdb* seja:

:deny

telles@10.7.8.9:allow

192.168.0.10:allow,ALLOWED="Yes"

Ex:

\$ export TCPREMOTEINFO="telles"

\$ tcprulescheck cdb

rule :

deny connection

\$ export TCPREMOTEIP="10.7.8.9"

\$ tcprulescheck cdb

rule telles@10.7.8.9:

allow connection

\$ export TCPREMOTEIP="192.168.0.10"

\$ tcprulescheck cdb

rule 192.168.0.10:

set environment variable ALLOWED=Yes

allow connection

tcpserver

tcpserver [-1UXpPhHrRoOdDqQv] [-c n] [-x regras.cdb] [-B banner] [-g gid] [-u uid] [-b n] [-l nomelo-cal] [-t n] host porta programa

Aceita conexões TCP de entrada.

tcpserver aguarda por conexões TCP de clientes. Para cada conexão, executa *programa* recebendo informações de rede via descritor 0 e enviando via descritor 1. Por padrão, cria uma série de variáveis TCP.

tcpserver será finalizado se receber um sinal kill do tipo *SIGTERM*.

Opções	Descrição
-q	Não imprime mensagens de erro na tela.
-Q	Imprime mensagens de erro na tela (Padrão).
-v	Imprime mensagens de erro e de status na tela.
-c n	Número máximo de conexões simultâneas que serão aceitas. Deve ser um número positivo. (Padrão: 40).

Opções	Descrição
<code>-x regras.cdb</code>	Utiliza regras compiladas por <i>tcprules</i> no arquivo <i>regras.cdb</i> para setar variáveis de ambiente ou rejeitar conexões remotas. É possível executar <i>tcprules</i> para atualizar <i>regras.cdb</i> enquanto <i>tcpserver</i> estiver sendo executado.
<code>-X</code>	Utilizada em conjunto com <code>-x regras.cdb</code> , faz com que <i>tcpserver</i> aceite conexões caso <i>regras.cdb</i> não exista. Normalmente <i>tcpserver</i> rejeita conexões se <i>regras.cdb</i> não existir.
<code>-B banner</code>	Imprime banner na tela assim que a conexão é efetuada. <i>tcpserver</i> mostra o banner antes de verificar as variáveis de ambiente <i>\$TCPREMOTEHOST</i> e <i>\$TCPREMOTEINFO</i> e antes de checar <i>regras.cdb</i> . Este recurso pode ser utilizado para reduzir a latência em protocolos aonde o cliente aguarda um cumprimento do servidor.
<code>-g gid</code>	Altera o id de grupo do programa para <i>gid</i> depois de se preparar para receber a conexão. <i>gid</i> precisa ser um inteiro positivo.
<code>-u uid</code>	Altera o id de usuário do programa para <i>uid</i> depois de se preparar para receber a conexão. <i>uid</i> precisa ser um inteiro positivo.
<code>-U</code>	O mesmo que <code>-u \$UID</code> e <code>-g \$GID</code> . Tipicamente <i>\$UID</i> e <i>\$GID</i> são setados utilizando <i>envuidgid</i> .
<code>-l</code>	Após preparar a recepção da conexão, imprime na tela a porta local utilizada na conexão.
<code>-b n</code>	Permite acumular aproximadamente <i>n</i> conexões TCP do tipo SYN (backlog). Em vários sistemas, <i>n</i> é silenciosamente limitado em 5. Em sistemas que suportam <i>SYN cookies</i> , o backlog é irrelevante.
<code>-o</code>	Não altera opções do protocolo IP nos pacotes da conexão.
<code>-O</code>	Deleta opções do protocolo IP nos pacotes da conexão (Padrão).
<code>-d</code>	Atrasa o envio de dados em uma fração de segundos caso o host remoto responda lentamente (Padrão).
<code>-D</code>	Nunca atrasa o envio de dados. Habilita a opção <i>TCP_NODELAY</i> .
<code>-h</code>	Tenta obter o nome do host remoto via DNS para setar a variável de ambiente <i>\$TCPREMOTEHOST</i> (Padrão).
<code>-H</code>	Não tenta obter o nome do host remoto via DNS e remove a variável de ambiente <i>\$TCPREMOTEHOST</i> caso esteja setada. Para evitar loops, utilize esta opção para servidores rodando na porta 53.
<code>-p</code>	Paranoid. Depois de obter o nome do host remoto via DNS, tenta obter o endereço IP deste host via DNS; remove a variável de ambiente <i>\$TCPREMOTEHOST</i> se não encontrar o endereço IP para o nome do host ou se o endereço IP retornado não for o endereço IP do cliente.
<code>-P</code>	Não utiliza Paranoid (Padrão).
<code>-l nomelocal</code>	Não tenta resolver <i>nomelocal</i> em DNS. Utiliza <i>nomelocal</i> para a variável de ambiente <i>\$TCPLOCALHOST</i> . Recomendado utilizar 0 (zero) como <i>nomelocal</i> . Para evitar loops, utilize esta opção para servidores rodando na porta 53.

Opções	Descrição
-r	Tenta obter informações do host remoto para setar a variável de ambiente <i>\$TCPREMOTEINFO</i> (Padrão).
-R	Não tenta obter informações do host remoto. Para evitar loops, utilize esta opção para servidores rodando na porta 53 e 113.
-t <i>n</i>	Desiste de obter informações do host remoto para setar <i>\$TCPREMOTEINFO</i> após <i>n</i> segundos (timeout – Padrão: 26 segundos).
host	O endereço IP local do servidor será obtido do nome de host <i>host</i> . Se for utilizado 0 (zero), tcpserver permitirá que qualquer IP local receba conexões. <i>host</i> também pode ser um endereço IP do tipo <i>a.b.c.d</i> .
porta	Porta TCP local a ser utilizada. Deve ser um número ou um nome listado no arquivo <i>/etc/services</i> . Se for 0 (zero), tcpserver escolhe uma porta TCP livre.
programa	Programa a ser executado em caso de sucesso da conexão TCP.

who@

who@ *host-remoto*

Imprime a lista de usuários conectados em *host-remoto*.

Conecta em *host-remoto* na porta 11/TCP (systat) e imprime uma lista de usuários que estão conectados atualmente no sistema.

Remove o *CR* (carriage return - \015) da entrada recebida e converte caracteres que não podem ser impressos na tela em um formato visível.

Se *host-remoto* não for informado, who@ irá conectar em *localhost*.

Alguns computadores respondem na porta 11/TCP com uma lista de usuários atualmente conectados.

Ex:

```
# tcpserver -RHl0 0 11 /usr/bin/who
```

```
$ who@
```

```
telles pts/1 Aug 6 16:29
```

```
gilson pts/2 Aug 6 16:29
```

```
cleri pts/3 Aug 6 16:29
```

```
ana pts/4 Aug 6 16:29
```

Variáveis de ambiente utilizadas por tcp-env, tcp-server e tcpclient

Opções	Descrição
\$PROTO	A string TCP.
\$TCPLOCALHOST	O domínio do host local com letras maiúsculas convertidas em minúsculas. Caso nenhum domínio esteja disponível para o IP local, <i>\$TCPLOCALHOST</i> não será setado.
\$TCPLOCALIP	O IP local do host no formato <i>a.b.c.d</i> .
\$TCPLOCALPORT	A porta TCP do host local no formato decimal.
\$TCPREMOTEHOST	O domínio do host remoto com letras maiúsculas convertidas em minúsculas. Caso nenhum domínio esteja disponível para o IP remoto, <i>\$TCPREMOTEHOST</i> não será setado.
\$TCPREMOTEINFO	Uma string específica da conexão, talvez o nome do usuário fornecido pelo host remoto via 931/1413/IDENT/TAP. Se o host remoto não fornecer informações sobre a conexão, <i>\$TCPREMOTEINFO</i> não será setado.
\$TCPREMOTEIP	O IP do host remoto.
\$TCPREMOTEPORT	A porta TCP do host remoto.

Programas do pacote daemontools

daemontools é um conjunto de ferramentas utilizadas para o gerenciamento de serviços e programas em sistemas UNIX e Unix-like.

envdir

envdir *diretório programa*

Cria variáveis de ambiente baseado em arquivos contidos em *diretório*, e executa *programa*.

envdir lê todos os arquivos contidos em *diretório* e seta variáveis de ambiente aonde, *NO-ME_DO_ARQUIVO=SEU_CONTEÚDO*.

Ex:

```
$ cat script.sh
```

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

```
echo "Guia de $TIPO sobre $ASSUNTO."
```

```
$ cat variaveis/TIPO
```

```
consulta
```

```
$ cat variaveis/ASSUNTO
```

```
qmail
```

\$ envdir ./variaveis ./script.sh

Guia de consulta sobre qmail.

Se *NOME_DO_ARQUIVO* já estiver setado como variável de ambiente e o arquivo não estiver vazio, envdir remove a variável e seta novamente com o conteúdo de *NOME_DO_ARQUIVO*.

O nome do arquivo não pode conter o sinal “=” (igualdade); espaços em branco e TABS são removidos do final de linha do arquivo.

Se *NOME_DO_ARQUIVO* já estiver setado como variável de ambiente e o arquivo estiver vazio, envdir remove a variável e não seta novamente.

Caracteres nulos são substituídos por *new lines* em variáveis de ambiente.

Se não conseguir ler o conteúdo de *diretório*, não puder executar *programa* ou se ocorrerem problemas de memória que o impeçam de interagir com as variáveis de ambiente, envdir sai com código de erro 111.

Se não ocorrer nenhum erro, envdir sai com o mesmo código de erro de *programa*.

envuidgid

envuidgid *usuario programa*

Executa *programa* utilizando as variáveis de ambiente *\$UID* e *\$GID* da conta *usuario*.

Se não encontrar a conta *usuario* no sistema, não puder executar *programa* ou se ocorrerem problemas de memória que o impeçam de interagir com as variáveis de ambiente, envuidgid sai com código de erro 111.

Se não ocorrer nenhum erro, envuidgid sai com o mesmo código de erro de *programa*.

fghack

fghack *programa*

Força *programa* a rodar em primeiro plano (foreground).

Alguns programas quando executados, são automaticamente jogados para segundo plano (background); fghack pode forçar tais programas a continuarem em primeiro plano.

multilog

multilog [*te*] [*n num*] [*s size*] [*+ -*] *expressão* [*= arquivo*] [*! processador*] *diretório*

Recebe uma sequência de linhas via *STDIN* (entrada padrão) e faz log ou não de entradas que contenham *expressão* no arquivo “*current*” dentro de *diretório*.

Caso *expressão* não seja informada, todas as entradas serão logadas.

multilog sai com código de erro 0 (zero) quando encontra o final de *STDIN*. Se *STDIN* possuir um final de linha parcial, multilog adiciona um *new line* no final da linha.

multilog escreve uma mensagem de erro em *STDERR* (saída de erro padrão) e sai com código de erro 111, sem ler nenhuma entrada, se ocorrerem problemas de memória ou se outro processo multilog estiver escrevendo em um mesmo arquivo de log rotacionado por ele.

Quando executado no *Shell*, caracteres especiais devem ser escapados para não serem interpretados.

Se multilog não conseguir escrever em disco após começar a ler informações de *STDIN*, escreve uma mensagem de erro em *STDERR*, pausa e tenta novamente sem perder nenhuma informação.

NOTA: Isso pode tornar indisponível o programa que está enviando informações para multilog.

Opções	Descrição
t	Insere no início de cada linha de entrada, um timestamp preciso.
nnum	<i>num</i> será o número máximo de arquivos de log rotacionados que serão mantidos em <i>diretório</i> antes de iniciar a remoção do arquivo de log mais antigo. O rotacionamento de logs pode ocorrer quando <i>current</i> atingir o tamanho máximo de 99999 bytes ou quando o processo multilog anterior, recebeu um sinal <i>KILL</i> inesperado (diferente de <i>TERM</i>). O valor padrão é 10.
ssize	<i>size</i> é o tamanho máximo do arquivo de log em bytes. Após alcançar <i>size</i> , multilog fará o rotacionamento do arquivo. Seu novo nome terá o timestamp preciso do momento. <i>size</i> pode variar entre 4096 e 16777215 bytes. O valor padrão é 99999 bytes.
e	Faz multilog imprimir os primeiros 200 bytes de entrada na saída de erro padrão (stderr).
+ -expressão	Ação a ser tomada quando <i>expressão</i> for encontrada. Veja <i>Exemplo 4</i> .
=arquivo	Loga última entrada encontrada por expressão mais 1000 bytes em arquivo.
!processador	Programa/Ação a ser executado(a) quando multilog efetuar ações dentro de <i>diretório</i> (Ex: rotacionamento de logs). <i>processador</i> pode ser um programa contendo instruções a serem executadas na filtragem de <i>current</i> para ser armazenado em outro arquivo.

Exemplo 1:

```
$ while : ; do ( cat /tmp/input; sleep 5 ) ; done 2>&1 | multilog /logs &
```

```
[1] 2248
```

```
$ ls /logs
```

```
current lock state
```

```
$ cat /logs/current
```

```
cat: /tmp/input: No such file or directory
```

```
$ kill 2248
```

Se multilog receber um sinal kill do tipo *TERM*, somente terminará após identificar o final de linha da linha que estava sendo processada, garantindo assim, que nenhuma informação seja perdida.

Além do arquivo *current*, multilog criará os arquivos *lock* e *state* dentro de *diretório*, que serão utilizados para evitar que outro processo multilog escreva em *current*.

Exemplo 2:

```
$ while : ; do ( cat /tmp/input; sleep 5 ) ; done 2>&1 | multilog t /logs &
```

```
[1] 2321
```

```
$ cat /logs/current
```

```
cat: /tmp/input: No such file or directory
```

```
@40000000404a54df0749da34 cat: /tmp/input: No such file or directory
```

```
$ kill 2321
```

Com a utilização da opção “t”, multilog insere um timestamp preciso no início de cada linha do arquivo de log. *tail64nlocal* converte este formato de timestamp para um formato legível por humanos.

Exemplo 3:

```
$ while : ; do ( cat /tmp/input; sleep 5 ) ; done 2>&1 | multilog t n10 s1048576 /logs &
```

multilog manterá 10 (dez) arquivos de log em *diretório*, cada um com aproximadamente 1 MB de tamanho.

Após atingir 10 arquivos, multilog irá apagar o arquivo de timestamp mais antigo no próximo rotacionamento.

Exemplo 4:

```
$ while : ; do ( cat /tmp/input; sleep 5 ) ; done 2>&1 | multilog t '-*' '+*STATS:*' /logs &
```

A expressão “-*” instrui multilog a não selecionar nenhuma linha, ao mesmo tempo em que utiliza “+*STATS:*” para instruir multilog a selecionar entradas que contenham a expressão “STATS:” em qualquer parte da linha.

Desta forma, multilog vai logar em *current* apenas entradas que contenham a expressão “STATS:”.

Exemplo 5:

```
$ cat /tmp/script.sh
```

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

```
/usr/bin/tee -a ./acumulado.log
```

```
$ while : ; do ( cat /tmp/input; sleep 5 ) ; done 2>&1 | multilog t \!/tmp/script.sh /logs &
```

multilog passará todo o conteúdo do arquivo *current* pelo programa */tmp/script.sh* antes de executar o rotacionamento. Isso fará */tmp/script.sh* gerar/incrementar o arquivo *acumulado.log* com o conteúdo de todos os arquivos rotacionados.

O nome dos arquivos de log rotacionados por multilog, iniciam com um sinal de “@” seguidos de um timestamp preciso, finalizando com uma das seguintes extensões:

Extensão	Descrição
.s	O arquivo foi processado completamente e escrito de forma segura no disco.
.u	O arquivo não foi processado completamente durante uma interrupção inesperada, e possivelmente foi truncado no disco.

Quando multilog está rodando, seta a permissão de *current* para *644*; quando é finalizado de forma segura, seta a permissão de *current* para *744* restaurando a permissão para *644* quando for executado novamente.

pgrphack

pgrphack *programa*

Executa *programa* em um novo grupo de processos.

readproctitle

readproctitle *L D*

readproctitle recebe via *STDIN* (entrada padrão) informações de saída ou erros, e mantém um log na memória que é rotacionado automaticamente para ser visualizado pelo comando *ps*. readproctitle sempre mantém 3 (trez) pontos (...) à esquerda de *D*.

L consiste em uma série de argumentos e *D* é apenas um argumento, que deve ser no mínimo 5 (cinco) pontos (.....).

É comumente utilizado para suprimir saídas de erro de programas na console.

A quantidade de informações de saída ou de erros a ser exibida, depende do número de pontos (.) que foram utilizados em *D*, e será sempre preservado o final da informação.

Quando uma nova linha (new line) é encontrada, readproctitle substitue o *new line* por “!” (exclamação).

A maioria das ferramentas de listagem de processos (Ex: *ps*) substituem “!” por um sinal de “?” (interrogação) ou um “\n”.

readproctitle é finalizado quando encontra o final de *STDIN*.

Ex:

```
$ while : ; do ( cat /tmp/alo ; sleep 1 ) ; done 2>&1 | readproctitle saida/erro: ..... &
```

```
$ ps axw | grep readproctitle
```

```
2443 pts/1 S 0:00 readproctitle saida/erro: ... No such file or directory?
```

```
$ echo “Achei o arquivo” > /tmp/alo
```

```
$ ps axw | grep readproctitle
```

```
2443 pts/1 S 0:00 readproctitle saida/erro: ... directory?Achei o arquivo?
```

NOTA: O limite conhecido para o número máximo de pontos a serem utilizados é de 512 (512 bytes) e pode variar conforme o sistema operacional.

readproctitle está disponível a partir da versão 0.75 do pacote daemontools.

setlock

setlock [-nNxX] *arquivo programa*

Executa *programa* utilizando uma trava de arquivo (flock) para *arquivo*.

setlock cria *arquivo* caso não exista, e obtém uma trava exclusiva do arquivo.

Normalmente a trava de arquivo desaparece quando *programa* é finalizado.

Opção	Descrição
-n	Se arquivo estiver travado por outro processo, setlock abandona o processo.
-N	Se arquivo estiver travado por outro processo, setlock aguardará até que consiga obter uma nova trava para o arquivo (Padrão).
-x	Se arquivo não puder ser criado, aberto ou travado, setlock sai com código de erro 0 (zero).
-X	Se arquivo não puder ser criado, aberto ou travado, setlock imprime uma mensagem de erro e sai com código de erro diferente de 0 (Padrão).

setuidgid

setuidgid *usuario programa*

Executa *programa* utilizando *usuario* como usuário e grupo efetivos; remove qualquer grupo suplementar.

Se não encontrar a conta *usuario* no sistema, não conseguir alterar para *usuario* ou o grupo de *usuario* e não puder executar *programa*, setuidgid sai com código de erro 111.

Se não ocorrer nenhum erro, setuidgid sai com o mesmo código de erro de *programa*.

setuidgid não pode ser executado por nenhum outro usuário que não seja o usuário *root*.

softlimit

softlimit [**-mdslaopfct**] *programa*

Executa *programa* com novas limitações de recursos.

Opção	Descrição
-d <i>n</i>	Limita o segmento de dados por processo para <i>n</i> bytes.
-s <i>n</i>	Limita o segmento de pilha (stack) por processo para <i>n</i> bytes.
-l <i>n</i>	Limita as páginas físicas travadas por processo para <i>n</i> bytes. Não tem efeito na maioria dos sistemas operacionais.
-a <i>n</i>	Limita o total de todos os segmentos por processo para <i>n</i> bytes. Não tem efeito na maioria dos sistemas operacionais.
-m <i>n</i>	O mesmo que -d <i>n</i> -s <i>n</i> -l <i>n</i> -a <i>n</i> .
-o <i>n</i>	Limita o número de file descriptors por processo para <i>n</i> . Não tem efeito na maioria dos sistemas operacionais.

Opção	Descrição
-p <i>n</i>	Limita o número de processos por usuário para <i>n</i> .
-f <i>n</i>	Limita o tamanho do arquivo de saída para <i>n</i> bytes.
-c <i>n</i>	Limita o tamanho de arquivos do tipo <i>core</i> para <i>n</i> bytes.
-r <i>n</i>	Limita o RSS para <i>n</i> bytes. Este limite não é forçado a menos que a memória física esteja cheia.
-t <i>n</i>	Limita o tempo de CPU em <i>n</i> segundos. Este limite não é forçado a menos que o processo receba um sinal kil do tipo <i>SIGXCPU</i> após <i>n</i> segundos.

supervise

supervise *serviço*

Inicia e monitora um serviço. Vai para o diretório *serviço* e inicia o script *./run*.

Caso *./run* seja finalizado, supervise o inicia novamente. supervise pausa por 1 (hum) segundo após iniciar *./run*, evitando assim um possível loop na tentativa de iniciar *./run* por várias vezes seguidas.

Se o arquivo *serviço/down* existir, supervise não inicia *./run* imediatamente. É possível utilizar *svc* para iniciar *./run* e também para enviar outros comandos a supervise.

supervise mantém informações de status em formato binário dentro do arquivo *serviço/supervise*, o qual supervise precisa ter permissão de escrita. Esta informação de status pode ser lida por *svstat*.

supervise é finalizado imediatamente ao ser iniciado, se não encontrar os arquivos de que necessita dentro do diretório *serviço* ou se existir um outro processo supervise rodando no diretório *serviço*.

Uma vez que supervise for iniciado com sucesso, não finaliza enquanto não receber um sinal kill ou enquanto não receber uma solicitação de stop.

É possível utilizar *svok* para checar se supervise está rodando corretamente.

Utilize *svscan* para iniciar vários processos supervise de forma confiável.

SVC

svc [-udopchaitkx] *serviço*

Controla serviços monitorados por *supervise*.

Opção	Descrição
-u	Inicia o serviço se não estiver rodando. Reinicia se o serviço parar (Up).
-d	Se o serviço estiver rodando, envia um sinal <i>TERM</i> seguido de um sinal <i>CONT</i> . Depois de parado, não reinicia o serviço (Down).
-o	Inicia o serviço se não estiver rodando. Não reinicia se o serviço parar (Once).

Opção	Descrição
-p	Envia um sinal de <i>STOP</i> ao serviço (Pause).
-c	Envia um sinal de <i>CONT</i> ao serviço (Continue).
-h	Envia um sinal de <i>HUP</i> ao serviço (Hangup).
-a	Envia um sinal de <i>ALRM</i> ao serviço (Alarm).
-i	Envia um sinal de <i>INT</i> ao serviço (Interrupt).
-t	Envia um sinal de <i>TERM</i> ao serviço (Terminate).
-k	Envia um sinal de <i>KILL</i> ao serviço (Kill).
-x	supervise será finalizado assim que o serviço cair. Se você está usando esta opção em um sistema estável, está fazendo alguma coisa errada, pois supervise foi projetado para rodar eternamente.

svok

svok *serviço*

Checa se *supervise* está rodando corretamente no diretório *serviço*.

Sai com código de erro 0 (zero) se *supervise* estiver rodando corretamente, caso contrário sai com código de erro 100.

svscan

Inicia e monitora uma série de serviços.

Inicia um processo *supervise* para monitorar cada um dos serviços/subdiretórios contidos no diretório */service*.

Ignora qualquer subdiretório que inicie com “.” (ponto).

supervise precisa estar no *\$PATH* de *svscan*.

svscan opcionalmente inicia um par de processos *supervise*; um para o subdiretório (Ex: *serviço*) e outro para *serviço/log*, com um pipe entre eles. *svscan* age desta forma se, *serviço* não possuir mais que 255 bytes (255 caracteres) e se *serviço/log* existir e puder ser lido por *svscan* (Na versão 0.70 do daemontools e anteriores, *serviço* precisa ter sua flag *sticky* ligada para que *svscan* atue desta forma). *svscan* precisa de 2 (dois) descriptors livres para cada pipe.

A cada 5 (cinco) segundos, *svscan* checa os subdiretórios novamente. Se encontrar um novo subdiretório, inicia um novo processo *supervise*. Se encontrar um subdiretório antigo aonde seu processo *supervise* está parado, reinicia este processo *supervise*; no caso do subdiretório *serviço/log*, reutiliza o mesmo pipe, desta forma nenhum dado é perdido.

svscan foi projetado para rodar eternamente. Se houver problemas na criação de pipes ou com processos *supervise*, escreve uma mensagem de erro em *STDERR* e tenta novamente após 5 (cinco) segundos.

svscanboot

Inicia o programa *svscan* no diretório */service* e joga todas as mensagens de saída e de erro para o programa *readproctitle*.

svscanboot executa “*svscan /service 2>&1 | readproctitle service errors:*” seguido de 400 pontos, para tornar possível a visualização dos últimos 400 bytes de mensagens de erro que serão visíveis pelo comando *ps*, através de *readproctitle*.

svscanboot configura a variável de ambiente *\$PATH* para

“*/command:/usr/local/bin:/usr/local/sbin:/bin:/sbin:/usr/bin:/usr/sbin:/usr/X11R6/bin*”

e limpa todas as outras variáveis de ambiente.

NOTA: Disponível somente a partir da versão 0.75 do pacote *daemontools*.

svstat

svstat *serviço*

Imprime na tela o status do *serviço* que está sendo monitorado por *supervise*.

Ex:

svstat /service/qmail-smtpd

/service/qmail-smtpd: up (pid 1452) 729011 seconds

tai64n

Insere um timestamp preciso no início de cada linha recebida por *STDIN* (entrada padrão).

Para cada linha, imprime

1. um símbolo @,
2. um timestamp preciso,
3. um espaço em branco e
4. uma cópia da linha recebida

em *STDOUT*. O timestamp indica o momento que *tai64n* leu o primeiro caractere da linha de entrada.

Ex:

\$ echo "Processado com sucesso." | tai64n

@40000000404b6f9438f692ac Processado com sucesso.

\$ tai64n < status.log

@40000000404b70261d03c5c4 Programa saiu com erro 511.

tai64n sai com código de erro 0 (zero) quando encontra o final de *STDIN*; sai com código de erro 111 se houver problemas na leitura de *STDIN* ou escrita em *STDOUT*.

`tai64n` não aloca nenhuma memória do sistema depois de iniciado.

O timestamp utilizado por `tai64n` é um rótulo de 12 bytes do tipo *TAI64N*, que é impresso em 24 caracteres hexadecimais em minúsculo.

Ex:

4000000037c219bf2ef02e94

Este timestamp refere-se ao nanosegundo começando exatamente 935467455.787492500 segundos após o começo de 1970 TAI; o hexadecimal 37c219bf é 935467455 e 2ef02e94 é 787492500.

A implementação atual de `tai64n`, conta com a rotina da biblioteca *gettimeofday* do UNIX para retornar a hora corrente como o número de segundos TAI desde 1970-01-01 00:00:10 TAI. Deve-se tomar cuidado, pois a maioria das implementações de *gettimeofday* não estão de acordo com Y2038. Além do mais, a maioria dos relógios não são acertados corretamente.

Utilize *tai64nlocal* para converter este timestamp em um formato legível por humanos.

tai64nlocal

Recebe informações via *STDIN* (entrada padrão) e converte entradas de timestamp no formato *TAI64N* para um formato legível por humanos.

Se a linha recebida não iniciar com o caractere “@”, `tai64nlocal` escreve a linha em *STDOUT* sem nenhuma alteração. Se a linha iniciar com o caractere “@”, `tai64nlocal` verifica o timestamp logo após o “@” no formato impresso por *tai64n*, e escreve em *STDOUT* o timestamp convertido para a data local no formato ISO:

YYYY-MM-DD HH:MM:SS.SSSSSSSS.

Ex:

\$ echo "Processado com sucesso." | tai64n | tai64nlocal

2004-03-07 16:22:23.190959500 Processado com sucesso.

\$ tai64nlocal < Status.log

2004-03-07 16:23:35.752866500 Programa saiu com erro 511.

A implementação atual de `tai64nlocal`, conta com a rotina da biblioteca *localtime* do UNIX para encontrar a hora local.

Várias implementações de *localtime* utilizam uma escala de tempo incorreta, que não conta com o “pulo dos segundos”.

Deve-se tomar cuidado, pois a maioria das implementações de *localtime* não estão de acordo com Y2038.

`tai64nlocal` sai com código de erro 0 (zero) quando encontra o final de *STDIN*; sai com código de erro 111 se houver problemas na leitura de *STDIN* ou escrita em *STDOUT*.

`tai64nlocal` não aloca nenhuma memória do sistema depois de iniciado.