

Serviço de e-mail Postfix

Novembro/2017

Prof. Jairo

jairo@uni9.pro.br

professor@jairo.pro.br

<http://www.jairo.pro.br/>

Este material tem por única intenção reunir um conteúdo acadêmico necessário para auxiliar no ensino da disciplina "Prática e Administração de Sistemas Operacionais de Redes Livres", ministrado nos cursos de Tecnologias em Redes de Computadores e Segurança da Informação.

O conteúdo aqui exposto pode ser livremente redistribuído e usado como apoio de aula, desde que mantenha a sua integridade original.

O arquivo "postfix.pdf" pode ser livremente acessado em "http://www.jairo.pro.br/prat_adm_sist_oper/".

Qualquer crítica ou sugestão, favor entrar em contato com o Prof. Jairo no endereço eletrônico "jairo@uni9.pro.br" ou "professor@jairo.pro.br".

Sumário

1 - HISTÓRICO.....	3
2 - CONCEITOS SMTP.....	3
3 - CONCEITOS POP, IMAP, RELAY, SPAM E WEBMAIL.....	5
4 - INSTALAÇÃO DO SERVIÇO DE E-MAIL postfix.....	5
5 - CONFIGURAÇÃO DO SERVIÇO.....	8
6 - INICIALIZAÇÃO E TESTE DO SERVIÇO.....	9
7 - ENVIO DE E-MAIL DE UMA MÁQUINA PARA OUTRA.....	12
8 - LEITURA DE E-MAIL COM POP3 E IMAP.....	15
9 - MUA ALPINE.....	20

1 - HISTÓRICO

O correio eletrônico (e-mail) foi uma das primeiras aplicações práticas utilizadas no início da internet e, até hoje, é uma das mais utilizadas. Qualquer pessoa que possua um endereço eletrônico pode enviar e receber mensagens através da internet. O endereço conhecido como *e-mail address* ou endereço de correio eletrônico possui a estrutura básica **user@host.domain** ou **user@domain**, onde *user* representa o identificador do usuário, *host* representa o nome do *host* e *domain* é o nome do domínio na internet.

O funcionamento do correio eletrônico é baseado no paradigma "*store-and-forward*" (estocar e redirecionar), onde os usuários envolvidos na transferência de uma mensagem não interagem diretamente entre si, mas usam aplicações encarregadas de executar e gerenciar essa transferência.

2 - CONCEITOS SMTP

SMTP vem de *Simple Mail Transfer Protocol*, que é o protocolo usado na transferência de e-mails.

Os componentes principais de um sistema de correio eletrônico são:

- **Mail User Agent (MUA)** - aplicação que interage com o usuário, responsável pela obtenção de mensagens a serem transmitidas e também retirada de mensagens recebidas. São exemplos de MUA: Alpine¹, Evolution, Outlook, Thunderbird².
- **Mail Transport Agent (MTA)** - aplicação responsável pelo transporte de mensagens entre os pontos envolvidos, sejam elas na rede local ou através da internet. São exemplos de MTA: sendmail³, postfix⁴, qmail⁵, exim⁶, exchange.
- **Mail Delivery Agent (MDA)** - aplicação responsável pelo recebimento do e-mail entregue pelo MTA local, deposita o e-mail na caixa postal do usuário. O MDA tem a função básica de filtrar o conteúdo da mensagem. São exemplos de MDA: procmail e aplicações de filtragem contra vírus e spam em geral.
- **Mail Submission Agent (MSA)** - aplicação com funcionalidade semelhante ao MTA,

1 <http://www.washington.edu/alpine/>

2 <https://www.mozilla.org/en-US/thunderbird/>

3 <http://www.sendmail.org/>

4 <http://www.postfix.org/>

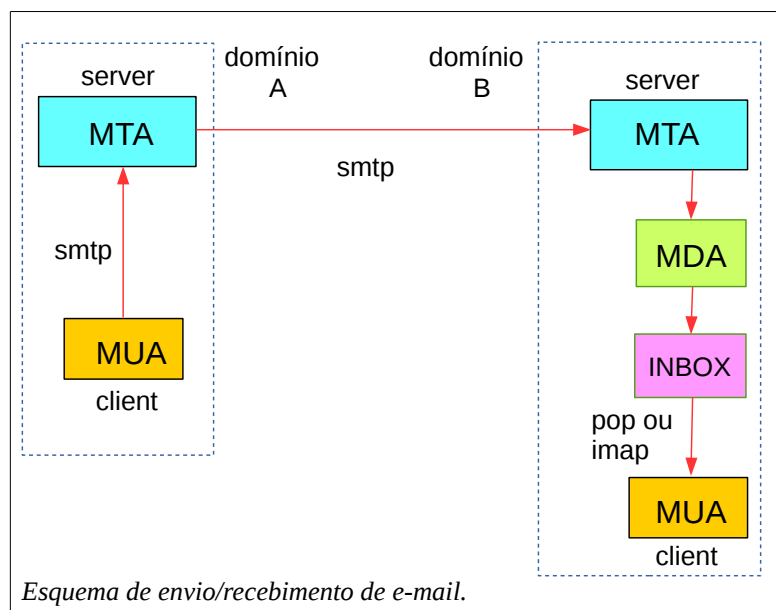
5 <http://www.qmail.org/>

6 <https://www.exim.org/>

porém normalmente somente aceita transportar e-mails na intranet (e-mails locais) e não de outros domínios (internet). Na prática, MSA é mais conceito que produto.

- **Mail Retrieval Agent/Mail Access Agent (MRA/MAA)** - aplicação usada pelo usuário para buscar seus e-mails depositados na Mail Box. Em muitos casos, aparece embutido (transparente) no MUA.
- **Mail Boxes** - caixas postais (inbox) onde são armazenadas as mensagens recebidas. O e-mail é depositado pelo MDA.

O princípio do funcionamento do correio eletrônico segue os seguintes passos: o remetente compõe a mensagem no seu aplicativo cliente de e-mail (agente de mensagens - MUA) e envia para o endereço eletrônico do destinatário. A mensagem é direcionada ao agente de transporte (MTA), cuja função é a entrega ou encaminhamento da mensagem à caixa postal correta do destinatário. Essa transferência é feita através do protocolo SMTP (*Simple Mail Transfer Protocol*). Na máquina destino, o MTA recebe a mensagem e a deposita na caixa postal correspondente do usuário.



Numa situação básica, o processo de envio/recebimento de e-mail pode ter três protocolos envolvidos:

- SMTP** (*Simple Mail Transfer Protocol*), responsável pelo envio de mensagens entre dois serviços de e-mail, que envolve pelo menos uma caixa postal de usuário (destinatário);
- POP** (*Post Office Protocol*) que gerencia o acesso às mensagens que chegaram à caixa postal de e-mail do usuário;
- IMAP** (*Internet Message Access Protocol*), que também gerencia o acesso às mensagens que chegaram à caixa postal de e-mail do usuário.

3 - CONCEITOS POP, IMAP, RELAY, SPAM E WEBMAIL

Existem três principais protocolos de estruturação de sistemas de e-mail: POP (*Post Office Protocol*, protocolo dos correios), DMSP (*Distributed Mail System Protocol*, protocolo distribuído do sistema do correio) e IMAP (*Internet Message Access Protocol*, protocolo de acesso de mensagem de internet).

Dos três, o POP é o mais antigo e conseqüentemente o mais conhecido. DMSP é limitado a uma única aplicação, PCMAIL, para sustentação da operação desconectada. O IMAP combina as potencialidades do POP e do DMSP, e fornece a melhor sustentação para todas as três modalidades do acesso remoto da caixa postal: *offline*, *online* e desconectado.

Outra vantagem do IMAP é a possibilidade de criar pastas (diretórios) no servidor de e-mail para poder guardar mensagens já lidas, e com isso desocupar espaço da caixa de entrada (*inbox*), além de permitir uma melhor organização das mensagens.

O *RELAY* é uma permissão que o serviço de e-mail concede ao usuário para poder enviar (*forward*) seus e-mails através desse MTA. Usualmente numa intranet, todos os usuários estão autorizados a usarem o MTA local, mas não é permitido o *RELAY* para usuários externos. A razão dessa limitação no uso é que, se for concedido acesso irrestrito para envio de e-mails, toda a internet irá pôr esse serviço MTA para transportar os e-mails deles, e muito dos quais passarão a enviar extensas listas de e-mail não solicitados com propaganda, que é o conceito de spam.

O Webmail permite a integração do serviço de e-mail com o serviço web, que é uma forma prática, simples, eficiente e segura de enviar/receber e-mails. Neste caso é o serviço web que invoca o serviço IMAP para acessar a caixa postal do usuário para ler e-mail, e é também o serviço web que faz relay no MTA para envio de mensagens.

A porta TCP padrão do serviço POP é 110, do serviço IMAP é 143, e do SMTP é 25. O SMTP também pode executar na porta 587, que é chamado de *submission*⁷. Se forem usados os protocolos seguros, pop3s é 995 e imaps é 993.

4 - INSTALAÇÃO DO SERVIÇO DE E-MAIL POSTFIX

Essa instalação também será *standalone*.

⁷ *Submission*: é um esquema de envio de e-mail que envolve a adição de um MSA (*Mail Submission Agent* - agente de submissão de e-mail) para receber as mensagens enviadas pelo MUA e repassar para o MTA. O objetivo do MSA é evitar que o MUA submeta a mensagem diretamente para o MTA. O uso de MSA é recomendável quando o MUA estiver numa rede distinta da do MTA. Um dos objetivos do MSA é combater o envio de SPAM. MSA também não é um protocolo seguro.

Num Ubuntu 12 (SystemV), para descobrir se o serviço **postfix** está instalado, procurar pelo seu script de inicialização em **/etc/init.d**:

```
root# ls /etc/init.d | grep postfix
/etc/init.d/postfix
```

NOTA 1: num sistema Ubuntu ou Debian SystemD, o comando equivalente seria "**systemctl list-units --all | grep postfix**";

NOTA 2: tanto num Ubuntu quanto num Debian, uma alternativa seria buscar pelo pacote de instalação, com o comando **dpkg -l | grep -i postfix**;

NOTA 3: tanto num CentOS quanto Red Hat SystemV, o o script de inicialização tem o mesmo nome, **/etc/init.d/postfix**;

NOTA 4: num CentOS ou Red Hat SystemD, o comando para saber se o serviço postfix está instalado é "**systemctl list-unit-files | grep postfix**";

NOTA 5: num CentOS ou Red Hat, para saber se o serviço postfix está instalado, uma alternativa seria procurar pelo pacote, com o comando "**rpm -aq | grep postfix**".

Se não houver saída no comando acima é indicativo de que o serviço **postfix** não está instalado. Nesse caso, instalar com o comando **apt-get**:

```
root# apt-get install postfix
```

NOTA 1: num CentOS ou Red Hat, o comando equivalente para instalar o serviço postfix é "**yum install postfix**".

NOTA 2: a instalação num Ubuntu ou Debian abre uma tela gráfica, onde é necessário usar a tecla <TAB> para navegar entre os menus para personalizar a instalação. Para o nosso caso, é melhor manter o *default* do instalador, que é "*Internet Site*".

5 - CONFIGURAÇÃO DO SERVIÇO

Primeiro vamos apenas olhar as configurações de instalação do serviço. Para isso, entrar no diretório `/etc/postfix`:

```
root# cd /etc/postfix
root# ls
dynamicmaps.cf  main.cf  master.cf  post-install  postfix-files  postfix-script  sasl
```

Existem dois arquivos de configuração, **master.cf** e **main.cf**. Vamos dar uma olhada no arquivo de configuração "**main.cf**" com o comando **more**:

```
root# more main.cf
```

Neste arquivo, uma configuração importante é **inet_interfaces**, que pela escolha de instalação "*Internet Site*" configurou para **inet_interfaces = all**. Mas se estivesse **inet_interfaces = localhost**, isto significaria que esse MTA somente aceitaria relay da interface de rede local (localhost), e desse modo não atenderia a nenhum cliente pela rede.

Com a configuração para "**inet_interfaces = all**", passa a aceitar relay de e-mail de qualquer origem na rede.

NOTA: o invés de **all**, o correto seria definir apenas a interface de rede pela qual os clientes acessam o serviço para o envio de e-mail.

Além disso, precisa também incluir o domínio **aluno.br** para poder enviar e-mail para "aluno@aluno.br". Precisa também declarar a rede de onde o MTA aceita *relay* de e-mails, que no nosso caso é 192.168.1.0/24.

Todas estas configurações são mostradas abaixo:

```
==== main.cf =====
myorigin = aluno.br
mydestination = $myhostname, localhost.$mydomain, localhost, aluno.br
mynetworks = 192.168.1.0/24, 127.0.0.0/8
inet_interfaces = all
=====
```


Em negrito estão as configurações que foram feitas no arquivo **main.cf**.

NOTA: se fosse o caso de habilitar também o serviço na porta **587**, que é chamado de **submission**, bastaria tirar o comentário da linha: "**submission inet n - - - - smtpd**".

6 - INICIALIZAÇÃO E TESTE DO SERVIÇO

Para verificar se o *daemon* está rodando, umas das maneiras é usar o comando **ps**:

```
root# ps -ef | grep postfix
```

Se houver saída no comando acima, isto indica que o serviço está rodando. Se este for o caso, parar o serviço neste momento.

Para parar o serviço num sistema SystemV, comandar:

```
root# /etc/init.d/postfix stop
Parando o postfix: [ OK ]
```

NOTA 1: num Ubuntu ou Debian SystemD, usar o mesmo comando acima;

NOTA 2: num CentOS ou Red Hat SystemD, usar o comando "**systemctl stop postfix**".

Uma vez confirmado que o serviço está parado, iniciar para tomar efeito as configurações feitas acima. Para iniciar num sistema SystemV, usar o script de inicialização:

```
root# /etc/init.d/postfix start
Iniciando o postfix: [ OK ]
```

NOTA 1: num Ubuntu ou Debian SystemD, usar o mesmo comando acima;

NOTA 2: num CentOS ou Red Hat SystemD, usar o comando "**systemctl start postfix**".

Agora, o comando **ps -ef | grep postfix** deverá mostrar que o serviço postfix está rodando.

Depois disso, o scan de portas com **nmap** vai mostrar que a porta 25 também está aberta:

```
root# nmap localhost

Starting Nmap 5.21 ( http://nmap.org ) at 2017-07-23 13:21 BRST
Interesting ports on localhost (127.0.0.1):
Not shown: 998 closed ports
PORT      STATE      SERVICE
25/tcp    open       smtp
631/tcp   open       ipp

Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```

Nesse ponto, já pode ser testado o envio de e-mail para o serviço de e-mail em *localhost*. O e-mail será enviado para o usuário local *aluno*, com e-mail **aluno@aluno.br**. Ou seja, o usuário *aluno* em *localhost* vai enviar um e-mail para a caixa postal dele mesmo em *localhost*.

Como já sabemos, um MTA é um dispositivo que transporta tráfego SMTP. Neste processo, o cliente de envio MTA conecta no MTA que vai receber a mensagem, que assim é transportada de um MTA para o outro.

No nosso caso, só temos um MTA, que é o que vai receber a mensagem, então precisamos também de um agente que conecte nesse MTA, para enviar a mensagem. A função de agente de envio pode ser feita usando o comando **telnet**, para conectar na **porta 25** em *localhost* e enviar a mensagem:

```

root# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 ubuntu12 ESMTP Postfix (Ubuntu)
helo Postfix
250 localhost
Mail from: aluno@aluno.br
250 2.1.0 aluno@aluno.br... Sender ok
Rcpt to: aluno@aluno.br
250 2.1.5 aluno@aluno.br... Recipient ok
data
354 Enter mail, end with "." on a line by itself
Subject: teste
Apenas um teste de envio de e-mail.
.
250 2.0.0 q7QHCaPV004001 Message accepted for delivery
quit
221 2.0.0 Bye
Connection closed by foreign host.

```

Repare que a aplicação cliente **telnet** conectou na porta 25, que é onde está o serviço de e-mail. Na cor azul aparecem as respostas do MTA Postfix em *localhost*. Uma vez conectado, tem que especificar quem é o destinatário (**Rcpt to:** *aluno@aluno.br*), quem está originando (enviando) o e-mail (**Mail from:** *aluno@aluno.br*) e, só depois disso, usar a palavra *data* para iniciar o conteúdo da mensagem. Para finalizar a mensagem, digitar um ponto "." numa linha nova, comandar <enter> e depois escrever *quit*.

Esta mensagem foi para a caixa postal (INBOX) do usuário aluno, em **/var/spool/mail/aluno**. Para visualizar o conteúdo da mensagem recebida, basta usar o comando **tail**:

```

root# tail /var/spool/mail/aluno
From aluno@aluno.br Sun Jul 23 13:34:03 2017
Return-Path: <aluno@aluno.br>
Received: from localhost [127.0.0.1]
    by localhost (2.9.6/2.9.6) with SMTP id q7QHCaPV004001
    for aluno@aluno.br; Sun, 23 Jul 2017 13:34:03 -0300
Subject: teste
Message-Id: <201208261713.q7QHCaPV004001@localhost>
Date: Sun, 23 Jul 2017 13:34:06 -0300
From: aluno@aluno.br
Apenas um teste de envio de e-mail.

```

O arquivo de log do serviço de e-mail, **/var/log/mail.log**, vai acusar que a mensagem foi

enviada. Para visualizar este log, usar o comando **tail**:

```
root# tail /var/log/mail.log
Jul 23 13:34:03 127.0.0.1 postfix[4001]: q7QHCaPV004001: from=aluno@aluno.br, size=32, class=0,
nrcpts=1, msgid=<201208261713.q7QHCaPV004001@localhost>, proto=SMTP, daemon=MTA,
relay=localhost [127.0.0.1]
Jul 23 13:34:03 127.0.0.1 postfix[4015]: q7QHCaPV004001: to=aluno@aluno.br,
ctladdr=aluno@aluno.br (1000/1000), delay=00:00:30, xdelay=00:00:00, mailer=local, pri=30313,
dsn=2.0.0, stat=Sent
```

NOTA 1: num Ubuntu ou Debian, o arquivo de log é **/var/log/mail.log**.

NOTA 2: num CentOS ou Red Hat, o arquivo de log é **/var/log/maillog**.

7 - ENVIO DE E-MAIL DE UMA MÁQUINA PARA OUTRA

O teste agora consiste em enviar um e-mail de uma máquina para outra. No endereço IP 192.168.1.11 está o cliente, e no endereço 192.168.1.10 está o MTA que vai receber a mensagem.

Mas para haver transporte de mensagens de uma máquina para a outra, precisaria um registro MX no serviço de nomes, pois o endereço e-mail é do tipo **user@domain** ou **user@host.domain**.

Então, para este envio, seria necessário primeiro configurar um serviço de nomes e declarar num registro MX que o host em 192.168.1.10 hospeda o serviço de e-mail. Porém, isso daria algum trabalho adicional e tiraria o foco, que é apenas testar o envio de e-mail.

Uma alternativa ao serviço de e-mail com registro MX é declarar no arquivo **/etc/hosts** das duas máquinas os seus respectivos nomes (*hostnames*) e endereços IP. Deste modo, mesmo sem serviço de nomes entre estas duas máquinas, poderá ser usado nomes dos *hosts*, pois os arquivos **/etc/hosts** traduzem esses nomes para endereços IP.

Vamos agora declarar os *hostnames* e endereços IP destas duas máquinas em seus respectivos arquivos **/etc/hosts**.

O formato no arquivo **/etc/hosts** consiste de entradas para endereços IPs e *hostnames*, deste modo basta declarar neste arquivo o endereço IP e o *hostname* da máquina que hospeda o serviço MTA, e também o *hostname* e o endereço IP da máquina cliente.

E para saber qual é o nome do host, basta usar o comando **hostname**:

```
root# hostname
host10
```

NOTA 1: não necessariamente precisa declarar exatamente o *hostname* da máquina, pode ser declarado um apelido para a máquina;

NOTA 2: no arquivo **/etc/nsswitch.conf** é definida a ordem de procura pelo nome da máquina, que pode ser, por exemplo, procurar primeiro no **/etc/hosts** e depois no DNS.

Na máquina cliente, no endereço 192.168.1.11, o comando **hostname** retorna que seu nome é **host11**.

Então, nas máquinas 192.168.1.10 e 192.168.1.11, precisa editar o arquivo **/etc/hosts** e incluir as duas seguintes linhas de configuração:

```
===== /etc/hosts =====
192.168.1.10      host10
192.168.1.11      host11
=====
```

Após incluir as entradas no **/etc/hosts**, na máquina 192.168.1.10, testar com o comando **ping** para verificar se a máquina que hospeda o serviço MTA responde pelo nome **host10**, e também se o cliente responde pelo nome **host11**:

```
root# ping host10
PING host10 (192.168.1.10) 56(84) bytes of data.
64 bytes from host10 (192.168.1.10): icmp_seq=1 ttl=58 time=11.5 ms
64 bytes from host10 (192.168.1.10): icmp_seq=2 ttl=58 time=16.4 ms
^C
root# ping host11
PING host11 (192.168.1.11) 56(84) bytes of data.
64 bytes from host11 (192.168.1.11): icmp_seq=1 ttl=58 time=11.5 ms
64 bytes from host11 (192.168.1.11): icmp_seq=2 ttl=58 time=16.4 ms
^C
```

NOTA: para cessar a execução do ping, pressionar simultaneamente as teclas control "<ctrl>" e "c".

Na máquina cliente, 192.168.1.11, também deve ser testado se **host10** e **host11** respondem ao **ping**:

```

shell_cliente$ ping host10
PING host10 (192.168.1.10) 56(84) bytes of data.
64 bytes from host10 (192.168.1.10): icmp_seq=1 ttl=58 time=12.3 ms
64 bytes from host10 (192.168.1.10): icmp_seq=2 ttl=58 time=15.7 ms
^C
shell_cliente$ ping host11
PING host11 (192.168.1.11) 56(84) bytes of data.
64 bytes from host11 (192.168.1.11): icmp_seq=1 ttl=58 time=10.9 ms
64 bytes from host11 (192.168.1.11): icmp_seq=2 ttl=58 time=16.1 ms
^C

```

NOTA: se não responder ao **ping** pelo nome declarado no **/etc/hosts**, isso é devido à máquina estar inacessível ou a entrada no **/etc/hosts** não estar correta.

Conforme fizemos no exemplo anterior, para enviar o e-mail será usada uma aplicação MUA, mas sim conectar direto na máquina que hospeda o serviço MTA com **telnet** na porta 25. A diferença é que agora este acesso e envio do e-mail será feito pelo cliente **host11** no IP 192.168.1.11.

O e-mail será enviado pelo usuário aluno em **host11** para aluno no MTA em **host10**. No e-mail (*Mail from:*) é `aluno@aluno.br` e no (*Rcpt to:*) também é `aluno@aluno.br`.

```

shell_cliente$ telnet host10 25
Trying 192.168.1.10...
Connected to host10.
Escape character is '^]'.
220 host10 ESMTP Postfix
helo Postfix
250 host10
Mail from: aluno@aluno.br
250 2.1.0 Ok
Rcpt to: aluno@aluno.br
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: Outro teste
Outro teste, agora o e-mail é enviado de 192.168.1.11 para 192.168.1.10
.
250 2.0.0 Ok: queued as 533F98C09BE
quit
221 2.0.0 Bye
Connection closed by foreign host.

```

Na cor **azul** aparecem as respostas do MTA Postfix em **host10** (192.168.1.10).

Esta mensagem foi enviada para a caixa postal (INBOX) do usuário aluno, em `/var/spool/mail/aluno` na máquina MTA (**host10**, 192.168.1.10). Nesta máquina, para visualizar o conteúdo da mensagem recebida, basta usar o comando **tail**:

```
root# tail /var/spool/mail/aluno
From aluno@aluno.br Sun Jul 23 13:40:27 2017
Return-Path: <aluno@aluno.br>
X-Original-To: aluno@aluno.br
Delivered-To: aluno@aluno.br
Received: from host11 (host11 [192.168.1.11])
    by host10.localdomain (Postfix) with SMTP id 533F98C09BE
    for <aluno@aluno.br>; Sun, Jul 2017 13:40:29 -0300 (BRT)
Subject: Outro teste

Outro teste, agora o e-mail é enviado de 192.168.1.11 para 192.168.1.10
```

E o arquivo de log `/var/log/maillog`, na máquina que hospeda o serviço de e-mail (192.168.1.10), vai acusar que a mensagem foi recebida. Para visualizar este log, usar o comando **tail**:

```
root# tail /var/log/mail.log
Jul 23 13:40:29 host10 postfix/smtpd[9277]: 533F98C09BE:
client=host11[192.168.1.11]
Jul 23 13:40:29 host10 postfix/cleanup[9294]: 533F98C09BE: message-id=<>
Jul 23 13:40:29 host10 postfix/qmgr[9059]: 533F98C09BE: from=<aluno@aluno.br>,
size=269, nrcpt=1 (queue active)
Jul 23 13:40:29 pinguim postfix/local[9304]: 533F98C09BE: to=<aluno@aluno.br>,
relay=local, delay=148, delays=148/0.01/0/0.05, dsn=2.0.0, status=sent (delivered to
mailbox)
Jul 23 13:40:29 pinguim postfix/qmgr[9059]: 533F98C09BE: removed
Jul 23 13:40:29 pinguim postfix/smtpd[9277]: disconnect from host11[192.168.1.11]
```

NOTA 1: num Ubuntu ou Debian, o arquivo de log é `/var/log/mail.log`.

NOTA 2: num CentOS ou Red Hat, o arquivo de log é `/var/log/maillog`.

8 - LEITURA DE E-MAIL COM POP3 E IMAP

Uma vez o e-mail depositado na caixa de entrada do usuário (INBOX), o modo usual de

acessar estas mensagens é através do serviço pop3 ou imap.

Estes dois serviços costumam ser instalados juntos, pela instalação do pacote **dovecot**.

Num Ubuntu 12 (SystemV), para descobrir se o pacote **dovecot** está instalado, procurar pelo seu script de inicialização em **/etc/init.d**:

```
root# ls /etc/init.d | grep dovecot
/etc/init.d/dovecot
```

NOTA 1: também num Ubuntu ou Debian SystemD, procurar pelo script de inicialização. Embora o sistema seja SystemD, o serviço opera no modo SystemV;

NOTA 2: num CentoOS ou Red Hat SystemV, procurar pelo script de inicialização em **"/etc/init.d/dovecot"**;

NOTA 3: num CentOS ou Red Hat SystemD, procurar pelo arquivo de *unit*, com o comando **"systemctl list-unit-files | grep dovecot"**;

NOTA 4: em qualquer Ubuntu ou Debian, uma alternativa seria procurar pelo pacote instalado, com o comando **"dpkg -l | grep dovecot"**;

NOTA 5: em qualquer CentOS ou Red Hat, uma alternativa seria procurar pelo pacote instalado, com o comando **"rpm -aq | grep dovecot"**.

Se não houver saída no comando acima, isto é indicativo de que o serviço **dovecot** não está instalado. Nesse caso, num Ubuntu ou Debian instalar com o comando **apt-get**:

```
root# apt-get install dovecot-core dovecot-pop3d dovecot-imapd
```

NOTA: num CentOS ou Red Hat, o comando para instalar é **"yum install dovecot"**.

Depois de instalado, precisa configurar os arquivos **/etc/dovecot/dovecot.conf**, **/etc/dovecot/conf.d/10-auth.conf** e **/etc/dovecot/conf.d/10-mail.conf**.

Em **dovecot.conf** precisa incluir a seguinte linha:

```
==== /etc/dovecot/dovecot.conf =====
protocols = pop3 imap # não precisa incluir pop3s nem imaps, pois já são default
=====
```


Em **10-auth.conf**, precisa **descomentar e corrigir a seguinte linha**:

```
==== /etc/dovecot/conf.d/10-auth.conf =====
disable_plaintext_auth = no # para poder trabalhar com protocolos inseguros
=====
```

NOTA: no arquivo acima, a linha "**disable_plaintext_auth = yes**" está comentada, então precisa ser descomentada e alterada de yes para **no**, isso para aceitar também acesso na porta padrão do serviço (não seguro).

Em **10-mail.conf**, precisa **descomentar e corrigir a seguinte linha**:

```
==== /etc/dovecot/conf.d/10-mail.conf =====
mail_location = mbox:~/mail:INBOX=/var/spool/mail/%u # INBOX em /var/spool/mail
=====
```

NOTA: a linha original está "mail_location = mbox:~/mail:INBOX=/var/mail/%u", e precisa ser alterada para "mail_location = mbox:~/mail:INBOX=/var/**spool**/mail/%u", pois as caixas de e-mail estão no diretório **/var/spool/mail**.

Depois disso é necessário reiniciar o serviço dovecot. Num Ubuntu 12 (SystemV), comandar:

```
root# /etc/init.d/dovecot restart
```

NOTA 1: num Ubuntu ou Debian, o simples fato de instalar o pacote como o **apt-get** já inicia o serviço. Por isso foi dado um restart;

NOTA 2: num CentOS ou Red Hat SystemV, o fato de instalar com o **yum** não inicia o serviço, portanto agora iniciar com o comando "**/etc/init.d/dovecot start**";

NOTA 3: em qualquer sistema SystemD, iniciar o serviço com o comando "**systemctl start dovecot**", ou reiniciar com o comando "**systemctl restart dovecot**".

Agora, os serviços **pop3**, **pop3s**, **imap** e **imaps** devem estar rodando e com as suas respectivas portas de serviço abertas. Um scan com **nmap** vai mostrar:

```

root# nmap localhost

Starting Nmap 5.21 ( http://nmap.org ) at 2017-07-23 13:51 BRST
Interesting ports on localhost (127.0.0.1):
Not shown: 998 closed ports
PORT      STATE      SERVICE
25/tcp    open      smtp
110/tcp   open      pop3
143/tcp   open      imap
631/tcp   open      ipp
993/tcp   open      imaps
995/tcp   open      pop3s

Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds

```

Para testar o acesso nas portas padrão do pop3 e imap (serviços inseguros), usar o telnet na respectiva porta (110 e 143):

```

shell_cliente$ telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK Dovecot ready.
user aluno
+OK
pass uninove
+OK Logged in.
list
+OK 1 messages:
1 488
.
quit
+OK Logging out.
Connection closed by foreign host.

```

Na cor **azul** aparecem as respostas do serviço **pop3 dovecot**. Além dos comandos acima, o serviço **pop3** suporta também **retr**, **top** e **dele**. Por exemplo:

retr 1 => mostra a mensagem de número 1;
top 1 => mostra o topo (cabeçalho apenas) da mensagem de número 1;
dele 1 => exclui a mensagem de número 1.

Para testar o **imap**, acessar a porta 143 e seguir a sintaxe de comando deste protocolo:

```

shell_cliente$ telnet localhost 143
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE
STARTTLS AUTH=PLAIN] Dovecot ready.
a1 login aluno uninove
a1 OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE
SORT SORT=DISPLAY THREAD=REFERENCES THREAD=REFS THREAD=ORDEREDSUBJECT
MULTIAPPEND URL-PARTIAL CATENATE UNSELECT CHILDREN NAMESPACE UIDPLUS
LIST-EXTENDED I18NLEVEL=1 CONDSTORE QRESYNC ESEARCH ESORT SEARCHRES
WITHIN CONTEXT=SEARCH LIST-STATUS SPECIAL-USE BINARY MOVE] Logged in
a2 list "" "*"
* LIST (NoInferiors \UnMarked) "/" saved-messages
* LIST (NoInferiors \UnMarked) "/" sent-mail
* LIST (\HasNoChildren) "/" INBOX
a2 OK List completed.
a3 examine INBOX
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS ()] Read-only mailbox.
* 1 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1487080252] UIDs valid
* OK [UIDNEXT 2] Predicted next UID
a3 OK [READ-ONLY] Examine completed (0.000 secs).
a4 logout
* BYE Logging out
a4 OK Logout completed.
Connection closed by foreign host.

```

Na cor **azul** aparecem as respostas do serviço **imap Dovecot**. Além dos comandos acima, o serviço **imap** suporta também **fetch**, **create**, **copy** e **rename**, entre outros.

Para testar o acesso usando protocolo seguro, usar o comando **openssl**. Por exemplo, para acessar o serviço pop3s na porta 995:

```

shell_cliente$ openssl s_client -connect localhost:995
CONNECTED(00000003)
depth=0 OU = IMAP server, CN = imap.example.com, emailAddress = postmaster@example.com
verify error:num=18:self signed certificate
verify return:1
...
+OK Dovecot ready.
user aluno
+OK
pass uninove10
+OK Logged in.
quit
+OK Logging out.
closed

```

Depois de conectado, é só seguir a sintaxe do pop3. Ou do imap, caso tenha conectado na porta 993.

9 - MUA ALPINE

Mas para ler as mensagens de e-mail direto no servidor SMTP, uma ferramenta bem útil é o **alpine** (ou, simplesmente, pine). Para saber se está instalado, num Ubuntu ou Debian, comandar:

```
root# dpkg -l alpine
```

NOTA: num CentOS ou Red Hat, para checar se o pacote está instalado usar o comando "**rpm -aq | grep -i alpine**".

Caso não estivesse instalado, num Ubuntu ou Debian, bastaria comandar:

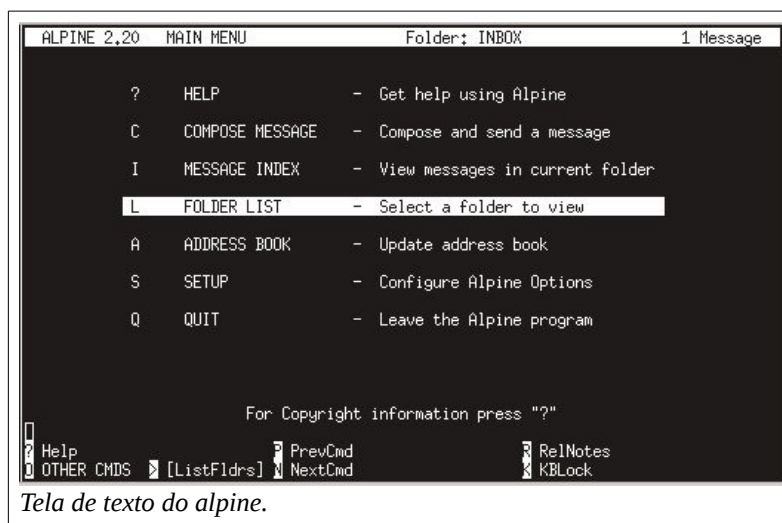
```
root# apt-get install alpine
```

NOTA: num CentOS ou Red Hat, o comando para instalar é "**yum install alpine**".

Para acessar as mensagens, comandar **alpine** como aluno:

```
shell_cliente$ id
uid=1001(aluno) gid=1001(aluno) groups=1001(aluno)
shell_cliente$ alpine
Creating subdirectory "/home/aluno/mail" where Alpine will store its mail folders.
```

Isso vai abrir a caixa de mensagens do usuário no shell:



```
ALPINE 2.20  MAIN MENU  Folder: INBOX  1 Message

?  HELP          - Get help using Alpine
C  COMPOSE MESSAGE - Compose and send a message
I  MESSAGE INDEX  - View messages in current folder
L  FOLDER LIST   - Select a folder to view
A  ADDRESS BOOK  - Update address book
S  SETUP         - Configure Alpine Options
Q  QUIT         - Leave the Alpine program

For Copyright information press "?"

? Help
OTHER CMDS [ListFldrs] PrevCmd RelNotes
NextCmd KBlock
```

Tela de texto do alpine.