

7. Correo electrónico y protocolos de correo

- El correo electrónico se reparte usando una arquitectura cliente/servidor.
- El correo se crea usando un programa cliente. Este programa envía el mensaje a un servidor. El servidor envía el mensaje al servidor de correo del receptor, donde el mensaje se pasa al cliente de correo del destinatario.
- Hay varios protocolos de red que permiten a máquinas con diferentes sistemas operativos intercambiar correo.
- Los protocolos de transporte manejan la comunicación entre la aplicación cliente y el servidor, y entre el servidor de origen al servidor de destino. El SMTP (Simple Mail Transfer Protocol) es el más frecuente.
- Los protocolos de acceso manejan la comunicación entre el cliente del receptor y el servidor del receptor. Los más frecuentes son POP (Post Office Protocol) e IMAP (Internet Message Access Protocol)

7. Protocolos de transporte de email: SMTP

- SMTP es el protocolo TCP/IP para repartir correo
- Usa el puerto 25
- Es un protocolo simple, se puede usar manualmente para comprobar si el SMTP de un equipo remoto está operativo
- SMTP no usa hosts intermedios para repartir el correo. Si el destinatario es un computador de sobremesa y no está encendido, el DNS proporciona mecanismos para enrutar un mensaje a un servidor de correo en vez de repartirlo directamente; el email se mueve del servidor al cliente cuando el cliente vuelve a estar online, y los protocolos utilizados para esta tarea son POP e IMAP.

7. SMTP

Comando	Sintaxis	Función
Hola	HELO <sending-host>	Identifica SMTP remitente
Remite	MAIL FROM:<direccion>	Dirección remite
Destinatario	RCPT TO:<to-address>	Dirección destino
Datos	DATA	Comienza mensaje
Reset	RSET	Aborta mensaje
Verifica	VRFT <string>	Verifica nombre usuario
Expande	EXPN <string>	Expande lista correo
Ayuda	HELP [string]	Ayuda
Salir	QUIT	Fin sesión SMTP

7. Servicios de correo: SMTP

220 Ready at Tue, 28 Jun 2004 17:21:26 EST

HELO pooh.example.com

250 Hello pooh.example.com, pleased to meet you

MAIL FROM:<daniel@pooh.example.com>

250 <daniel@pooh.example.com>... Sender ok

RCPT TO:<tyler@thoth.example.com>

250 <tyler@thoth.example.com>... Recipient ok

DATA

354 Enter mail, end with "." on a line by itself

Hi Tyler!

.

250 Mail accepted

QUIT

221 Delivering mail

- telnet mail.example.com 25

7. Protocolos de acceso: POP (Post Office Protocol)

- La versión más utilizada es POP3
- Es un protocolo de pregunta/respuesta tan simple como SMTP
- Usa el puerto 110

7. Comandos POP

Comando	Sintaxis	Función
Usuario	USER username	Nombre de usuario
Contraseña	PASS password	Contraseña
Estadísticas	STAT	No. mensajes sin leer
Recupera	RETR n	Recupera mensaje n
Elimina	DELE n	Elimina mensaje n
Ultimo	L*	Ultimo mensaje leído
Lista	LIST [n]	Tamaño de mensajes
Reset	RSET	Elimina borrado, no. = 1
Primero	TOP n l	Cabecera y n lines de m. n
No operación	NOOP	Nada
Salir	QUIT	Fin sesión POP

7. Servicios de correo: POP

```
+OK POP3 Server Process ready
USER hunt
+OK User name (hunt) ok. Password, please.
PASS Watts?Watt?
+OK 3 messages in folder NEWMAIL
STAT
+OK 3 459
RETR 1
+OK 146 octets
  The full text of message 1
.
DELE 1
+OK message # 1 deleted
RETR 2
+OK 155 octets
  The full text of message 2
.
DELE 2
+OK message # 2 deleted
RETR 3
+OK 158 octets
  The full text of message 3
.
DELE 3
+OK message # 3 deleted
QUIT

+OK POP3 Server exiting (0 NEWMAIL messages left)
```

- telnet mail.example.com 110

7. Protocolos de acceso: IMAP

- Alternativa a POP. Proporciona los mismos servicios y sincronización de carpetas
- IMAP4 usa el puerto 143.
- Es un protocolo que tiene un pequeño número de comandos pero es más complejo que POP y no es sencillo usarlo a mano (25 comandos)

7. Ejemplo IMAP

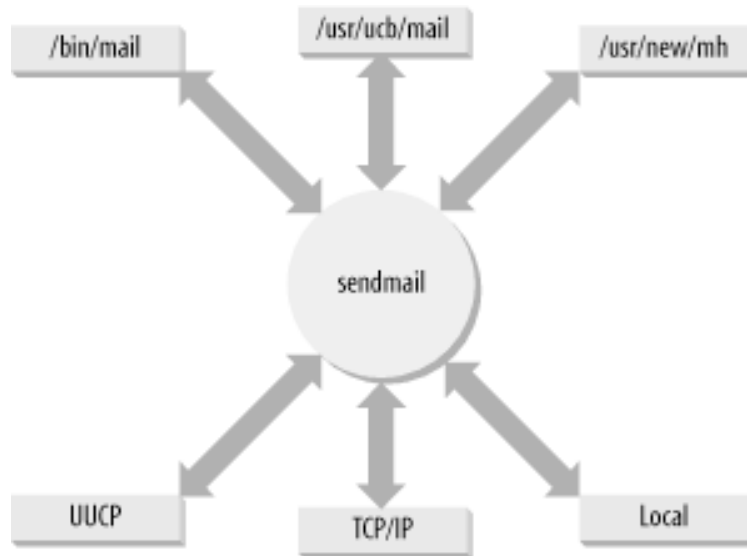
```
* OK mail.example.com IMAP4 server ready
a0001 LOGIN craig Wats?Watt?
a0001 OK LOGIN completed
a0002 SELECT inbox
* 3 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 965125671] UID validity status
* OK [UIDNEXT 5] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
* OK [UNSEEN 1]
a0002 OK [READ-WRITE] SELECT completed
a0003 FETCH 1 BODY[TEXT]
* 1 FETCH (BODY[TEXT] {1440}
... an email message that is 1440 bytes long ...
* 1 FETCH (FLAGS (\Seen))
a0003 OK FETCH completed
a0004 STORE 1 +FLAGS \DELETED
* 1 FETCH (FLAGS (\Seen \Deleted))
a0004 OK STORE completed
a0005 CLOSE
a0005 OK CLOSE completed
a0006 LOGOUT
* BYE mail.example.com IMAP4 server terminating connection
a0006 OK LOGOUT completed
```

telnet mail.example.com 143

7. Clasificación de programas de correo

- Todas las aplicaciones de correo caen en una de las siguientes tres categorías:
 - MTA (Mail Transport Agent). Un MTA transporta mensaje entre hosts usando SMTP. Un mensaje pueden utilizar varios MTAs hasta llegar a su destino. Sendmail, Postfix y Fetchmail son MTAs.
 - MDA (Mail Delivery Agent): un MDA mueve el correo desde el spool del MTA y el buzón del usuario. mail y Procmail son MDAs.
 - MUA (Mail User Agent): Programas basados en texto o gráficos para leer el correo, como mutt, evolution, thunderbird, etc.

7. MTA: Sendmail



- Sendmail sirve para enrutar el correo al usuario desde diferentes fuentes
- Acepta correo desde un programa de usuario, interpreta la dirección de destino, la reescribe en la forma adecuada para el programa de reparto y enruta el correo al programa adecuado.
- Con el correo entrante, interpreta la dirección y o bien entrega el mensaje al programa de correo del usuario o bien se lo reenvía a otro sistema.

7. Sendmail

- Adicionalmente, sendmail tiene otras dos funciones:
 - Recibe y entrega correo por medio de SMTP
 - Proporciona alias de direcciones de correo, lo que permite crear listas de distribución

7. Sendmail

- El daemon sendmail escucha en el puerto 25 y procesa el correo entrante. Por ejemplo, si se invoca mediante la orden

```
/usr/lib/sendmail -bd -q15m
```

se consigue que se consulte la cola con los mensajes que deben enviarse cada 15 minutos (-q15m) y que sendmail corra como un daemon y escuche en el puerto 25, lo cual sólo es necesario si el sistema acepta correo TCP/IP

- En general, las opciones de inicialización se manejan a través de un script de inicialización (/etc/rc.d/init.d/sendmail) y configuración en /etc/sysconfig/sendmail:

```
$ cat /etc/sysconfig/sendmail
DAEMON=yes
QUEUE=1h
```

7. Alias en sendmail

- Los alias sirven para proporcionar nombres alternativos para algunos usuarios, para reenviar el correo a otros equipos y para crear listas de distribución.
- Se definen en el fichero `/etc/aliases`
- El formato es `alias: recipient[, recipient,...]`

donde `alias` es el nombre del destinatario de correo, y `recipient` es el nombre al que el correo se entrega. `recipient` puede ser un nombre de usuario, otro alias o una dirección de email completa con nombre de usuario y de host. Pueden añadirse múltiples receptores a un mismo alias, creando una lista de distribución

7. Alias en sendmail

```
# special names
postmaster: clark
root: norman
# accept firstname.lastname@wrotethebook.com
rebecca.hunt: becky@rodent
jessie.mccafferty: jessie@jerboas
anthony.resnick: anthony@horseshoe
andy.wright: andy@ora
# a mailing list
admin: kathy, david@rodent, sara@horseshoe, becky@rodent, craig,
      anna@rodent, jane@rodent, christy@ora
owner-admin: admin-request
admin-request: craig
```

- Los dos primeros son nombres especiales (el correo dirigido a postmaster y a root se dirige a clark y norman)
- Los cuatro siguientes sirven para que el correo dirigido a un dominio llegue a la máquina adecuada
- Los restantes son una lista de distribución llamada “admin”

7. Alias en sendmail

- Sendmail no procesa el fichero aliases directamente, sino que se convierte en una base de datos mediante el comando newaliases o sendmail -bi
- Cada vez que se modifica el fichero aliases hay que ejecutar el comando newaliases

7. Reenvío de correo personal

- Sendmail permite que los usuarios reenvíen su correo a otra cuenta mediante el fichero `.forward`
- `.forward` se usa para reenvíos temporales, los definitivos se incluyen en alias
- El resto de la configuración de sendmail (todo menos alias y `.forward`) se define en el fichero `sendmail.cf`

7. sendmail.cf

- Las funciones de sendmail.cf son tres
 - Define el entorno de sendmail
 - Reescribe las direcciones con la sintaxis adecuada para el programa de recepción de correo
 - Asocia las direcciones con las instrucciones necesarias para entregar el correo
- La sintaxis de sendmail.cf es tan compleja que se usa un lenguaje de programación intermedio (m4) para crearlo

7. sendmail.cf

- Normalmente se parte de un ejemplo en m4 que se modifica y compila para crear un sendmail.cf. Por ejemplo,

```
divert(0)dnl
```

```
VERSIONID(`$Id: ch10.xml,v 1.7 2002/09/04 21:08:29 chodacki Exp $')
```

```
OSTYPE(`unknown')
```

```
FEATURE(`nouucp', `reject')
```

```
MAILER(`local')
```

```
MAILER(`smtp')
```

7. sendmail.cf

- divert(0)dnl: comandos que deben ser procesados. dnl elimina todo hasta el siguiente cambio de línea
- VERSIONID: control de versiones
- OSTYPE: configuración específica de operativo (bsd4.4, solaris, linux, etc.)
- FEATURE: en este caso, que el sistema no usa direcciones uucp
- MAILER: las últimas macros del fichero. Se especifican los mailers local y smtp. MAILER(local) especifica que se deben incluir las macros correspondientes al reparto de correo local entre usuarios de la máquina. MAILER(smtp) contiene las macros necesarias para enviar correo sobre una red TCP/IP

7. sendmail.cf

- Los mailers incluidos para TCP/IP son
 - smtp: 7bit
 - estmp: con MIME
 - smtp8: datos de 8 bit
 - dstmp: el sistema remoto accede a la cola de mensajes
 - relay: se reenvía el correo a otro servidor

7. Configuración

- Se copia el fichero patrón (por ejemplo, tcpproto.mc, visto en la transparencia anterior, a otro archivo temporal)
- Se reemplaza la palabra 'unknown' por linux, se guarda en linux.mc
- Se compila con m4: `m4 ../m4/cf.m4 linux.mc > sendmail.cf`
- Se copia el archivo resultante a etc
- Hay muchas opciones de configuración, que se verán en prácticas

7. SMTP en windows

- En Windows el servicio SMTP se instala con IIS, y existen sistemas propios de mensajería y correo electrónico, como Microsoft Exchange. También existen servicios POP3 e IMAP

7. MTA: Postfix

- **Postfix** es compatible con sendmail y más sencillo de configurar
- Postfix usa un diseño modular, donde pequeños procesos tienen privilegios limitados y son lanzados por un daemon maestro. Cada proceso realiza tareas específicas del reparto de correo y corren en un entorno restringido para limitar el efecto de los ataques.
- Los ficheros de configuración de postfix son legibles y no necesitan del procesador de macros de sendmail

7. Instalación por defecto de postfix

- El ejecutable es `/usr/sbin/postfix`. Los ficheros de configuración más usados de postfix son:
 - `access` - controla el acceso, los hosts autorizados a conectarse a postfix
 - `main.cf` - fichero global de configuración
 - `master.cf` - cómo postfix interactúa con los demás procesos para repartir correo
 - `transport` - asocia las direcciones de correo a los hosts que hacen relay
- El fichero `/etc/aliases` se comparte por postfix y sendmail

7. Configuración básica de postfix

- Por defecto, postfix no acepta conexiones externas. Para que se acepte correo de otras máquinas de la red, se edita **/etc/postfix/main.cf**
- Se quita el comentario (#) de la línea **mydomain**, se reemplaza **domain.tld** con el dominio del servidor de correo (p.e. example.com)
- Se quita el comentario de la línea **myorigin = \$mydomain**
- Se quita el comentario de la línea **myhostname** line, y se reemplaza **host.domain.tld** con el nombre de host de la máquina.
- Se quita el comentario de la línea **mydestination = \$myhostname, localhost.\$mydomain**
- Se quita el comentario de la línea **mynetworks** y se reemplaza **168.100.189.0/28** con la configuración de la red de los hosts que pueden conectarse a un servidor.
- Se quita el comentario de la línea **inet_interfaces = all**
- Se quita el comentario de la línea **inet_interfaces = localhost**
- Se reinicia el servicio **postfix** para que el host acepte mensajes hacia el exterior.

7. MDA, servidor POP e IMAP: Dovecot

/etc/dovecot/dovecot.conf

```
protocols = pop3 imap lmtp
```

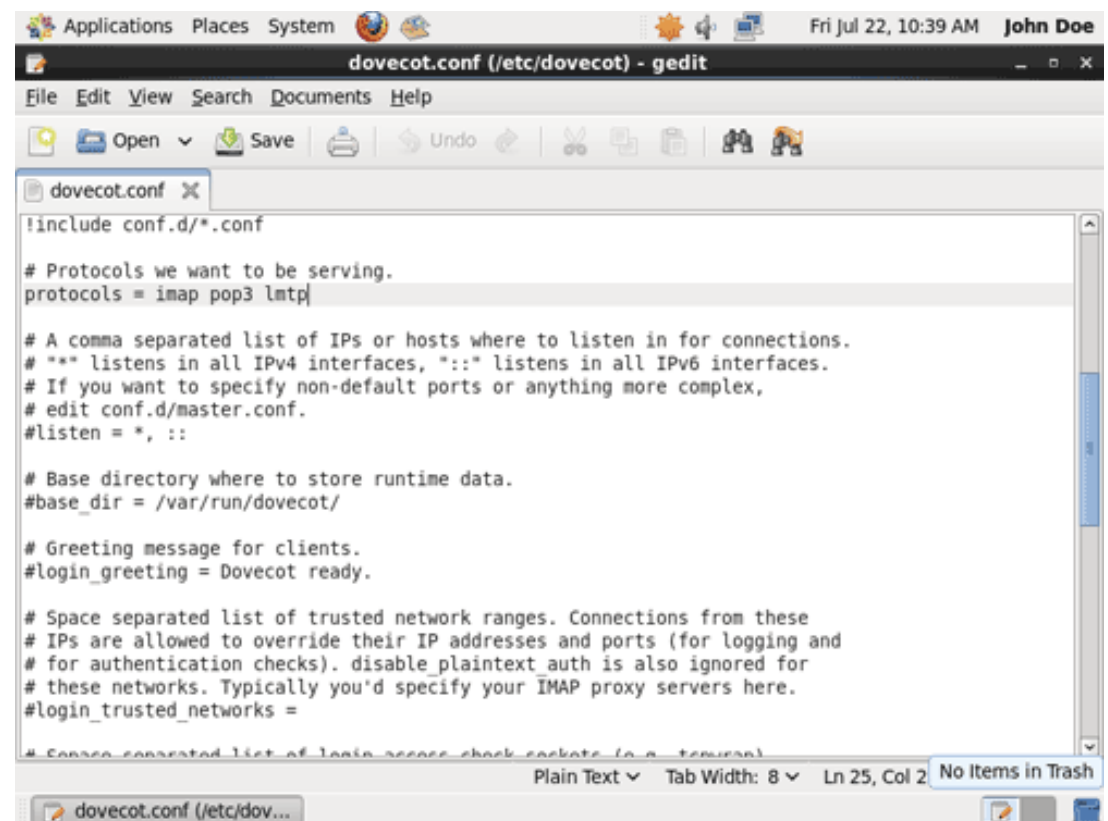
/etc/dovecot/conf.d/10-mail.conf

```
mail_location =  
maildir:~/Maildir
```

/etc/dovecot/conf.d/20-pop3.conf

```
pop3_uidl_format = %08Xu%08Xv
```

```
pop3_client_workarounds =  
outlook-no-nuls oe-ns-eoh
```



7. Test dovecot

```
[root@mail ~]# telnet localhost pop3
+OK dovecot ready.
user johndoe
+OK
pass password
+OK Logged in.
list
+OK 1 messages:
1 622
.
retr 1
+OK 622 octets
Return-Path: <johndoe@mail.acme.local>
X-Original-To: johndoe
Delivered-To: johndoe@mail.acme.local
Received: from localhost.localdomain (localhost.localdomain [127.0.0.1])
        by mail.acme.local (Postfix) with SMTP id 9729067C17
        for <johndoe>; Thu, 22 Feb 2007 09:06:37 -0500 (EST)
Message-Id: <20070222140640.9729067C17@mail.acme.local>
Date: Thu, 22 Feb 2007 09:06:37 -0500 (EST)
From: johndoe@mail.acme.local
To: undisclosed-recipients;;
X-IMAPbase: 1172153557 1
Status: 0
X-UID: 1
Content-Length: 5
X-Keywords:

test
.
quit
+OK Logging out.
Connection closed by foreign host.
[root@mail ~]#
```