

Utilidades básicas de redes

Así que finalmente se las ha arreglado para configurar su conexión de red, ¿y ahora? ¿cómo saber si está funcionando? ¿cómo saber si está configurada correctamente? ¿y qué hacer ahora que está configurada? Bien, este capítulo es para usted.

Herramientas de diagnóstico de red

Slackware Linux incluye un gran número de herramientas de red para resolución de problemas y diagnóstico de problemas de conexión de red, o simplemente para ver lo que hay por ahí en la red. La mayoría de estas herramientas son para que pueda ejecutarlas desde un terminal virtual o en una ventana de terminal en su escritorio gráfico. Algunas de ellas incluso tienen interfaces gráficas, pero vamos a tratar casi exclusivamente con herramientas de línea de comandos por ahora.

ping

ping(8) es una herramienta útil para determinar si un ordenador está operativo en su red o en Internet en general. Puede considerarlo un tipo de sonar para ordenadores. Al usarlo, usted envía un “ping” y escucha un eco para determinar si otro ordenador o dispositivo de red está escuchando. De manera predeterminada, **ping** verifica el equipo remoto una vez por segundo indefinidamente, pero puede modificar el intervalo entre verificaciones y el número total de comprobaciones fácilmente, sólo tiene que revisar la página de manual. Puede terminar la aplicación en cualquier momento con **CTRL+C**. Cuando **ping** ha terminado, muestra un práctico resumen de su actividad. **ping** es muy útil para determinar si un ordenador de su red o de Internet está disponible, pero algunos sistemas bloquean los paquetes que **ping** envía, así que a veces un sistema puede estar funcionando correctamente, pero no enviar respuestas.

```
darkstar:~# ping -c 3 www.slackware.com
64 bytes from slackware.com (64.57.102.34): icmp_seq=1 ttl=47 time=87.1 ms
64 bytes from slackware.com (64.57.102.34): icmp_seq=2 ttl=47 time=86.2 ms
64 bytes from slackware.com (64.57.102.34): icmp_seq=3 ttl=47 time=86.7 ms

--- slackware.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 86.282/86.718/87.127/0.345 ms
```

traceroute

traceroute(8) es una herramienta útil para determinar qué ruta toman sus paquetes para llegar a otra computadora. Es principalmente útil para determinar qué ordenadores están “cerca” o “lejos” de usted. Esta distancia no es estrictamente geográfica, ya que su proveedor de servicios de Internet puede enrutar el tráfico de su ordenador de forma extraña. **traceroute** le muestra cada uno de los enrutadores entre su ordenador y cualquier otra máquina a la que desee conectarse. Desafortunadamente, muchos proveedores, cortafuegos y enrutadores bloquearán **traceroute** así que puede que no obtengas un informe completo cuando lo use. Aún así, sigue siendo una

herramienta útil para la resolución de problemas de red.

```
darkstar:~# traceroute www.slackware.com
traceroute to slackware.com (64.57.102.34), 30 hops max, 46 byte
packets
 1gw.ctsmacon.com (192.168.1.254)1.468 ms2.045 ms1.387 ms
 210.0.0.1 (10.0.0.1)7.642 ms8.019 ms6.006 ms
 368.1.8.49 (68.1.8.49)10.446 ms9.739 ms7.003 ms
 468.1.8.69 (68.1.8.69)11.564 ms6.235 ms7.971 ms
 5dalsbbrj01-ae0.r2.dl.cox.net (68.1.0.142)43.859 ms43.287 ms 44.125 ms
 6dpr1-ge-2-0-0.dallasequinix.savvis.net (204.70.204.146)41.927 ms 58.247
ms44.989 ms
 7cr2-tengige0-7-5-0.dallas.savvis.net (204.70.196.29)42.577 ms 46.110
ms43.977 ms
 8cr1-pos-0-3-3-0.losangeles.savvis.net (204.70.194.53)78.070 ms 76.735
ms76.145 ms
 9bpr1-ge-3-0-0.LosAngeles.savvis.net (204.70.192.222)77.533 ms 108.328
ms120.096 ms
10wiltel-communications-group-inc.LosAngeles.savvis.net
(208.173.55.186)79.607 ms76.847 ms75.998 ms
11tg9-4.cr01.lsancarc.integra.net (209.63.113.57)84.789 ms85.436 ms85.575 ms
12tg13-1.cr01.sntdcabl.integra.net (209.63.113.106)87.608 ms 84.278 ms86.922
ms
13tg13-4.cr02.sntdcabl.integra.net (209.63.113.134)87.284 ms 85.924 ms86.102
ms
14tg13-1.cr02.rcrdcauu.integra.net (209.63.114.169)85.578 ms 85.285 ms84.148
ms
15209.63.99.166 (209.63.99.166)84.515 ms85.424 ms85.956 ms
16208.186.199.158 (208.186.199.158)86.557 ms85.822 ms86.072 ms
17sac-main.cwo.com (209.210.78.20)88.105 ms87.467 ms87.526 ms
18slackware.com (64.57.102.34)85.682 ms86.322 ms85.594 ms
```

telnet

En los días de antaño, **telnet**(1) era lo más grande desde el pan de molde. Básicamente, **telnet** abre una conexión de red no cifrada entre dos ordenadores y da el control de la sesión al usuario en lugar de a otra aplicación. Usando **telnet**, la gente podría conectarse a las shells en otros ordenadores y ejecutar comandos como si estuviese físicamente presente. Debido a su naturaleza no codificada, esto ya no es recomendable; sin embargo, **telnet** se sigue utilizando para este propósito por muchos dispositivos.

Hoy en día, **telnet** se utiliza mejor como una herramienta de diagnóstico de red. Al pasar el control de la sesión directamente al usuario, se puede utilizar para una gran variedad de propósitos de pruebas. Siempre y cuando sepa qué comandos ASCII enviar a la aplicación de la computadora, usted puede hacer cualquier número de actividades, tales como leer la web o comprobar su correo electrónico. Simplemente informe a **telnet** que puerto de red usar, y está todo listo.

```
darkstar:~$ telnet www.slackware.com 80
Trying 64.57.102.34...
```

```
Connected to www.slackware.com.  
Escape character is '^]'.  
HEAD / HTTP/1.1  
Host: www.slackware.com  
  
HTTP/1.1 200 OK  
Date: Thu, 04 Feb 2010 18:01:35 GMT  
Server: Apache/1.3.27 (Unix) PHP/4.3.1  
Last-Modified: Fri, 28 Aug 2009 01:30:27 GMT  
ETag: "61dc2-5374-4a973333"  
Accept-Ranges: bytes  
Content-Length: 21364  
Content-Type: text/html
```

ssh

Como hemos mencionado, **telnet** puede ser útil como herramienta de diagnóstico pero su naturaleza no encriptada lo convierte en un problema de seguridad para el acceso al shell. Afortunadamente, está el protocolo de shell seguro. Casi cada distribución de Linux, UNIX y BSD hoy en día utiliza OpenSSH, o **ssh**(1) para abreviar. Es una de las más importantes herramientas de red de uso común en la actualidad y hace uso de las más potentes técnicas criptográficas. **ssh** tiene muchas características, opciones de configuración, y hackeos limpios, suficientes para llenar su propio libro así que sólo entraremos en lo básico. Simplemente ejecute **ssh** con el nombre de usuario y el host y estarás conectado a él de forma rápida y segura. Si esta es la primera vez que está conectando a este ordenador, **ssh** le pedirá que confirme su deseo, y que haga una copia local de la clave de encriptación a utilizar. Si esta llave cambia más tarde, **ssh** le advertirá y se negará a conectarse porque es posible que alguien esté intentando secuestrar la conexión usando lo que se conoce como un ataque de hombre en el medio.

```
darkstar:~# ssh alan@slackware.com  
alan@slackware.com's password: secret  
alan@slackware.com:~$
```

El usuario y el nombre de host están en la misma forma que las direcciones de correo electrónico. Si no pone la parte del nombre de usuario, **ssh** utilizará su nombre de usuario actual al establecer la conexión.

tcpdump

Hasta ahora todas las herramientas que hemos visto se han centrado en hacer conexiones a otras computadoras, pero ahora vamos a ver el tráfico en sí. **tcpdump**(1) (que debe ejecutarse como root) nos permite ver todo o parte del tráfico de la red que se origina o se recibe por nuestro ordenador. **tcpdump** muestra los paquetes de datos en bruto de diversas maneras con todas las cabeceras de red intactas. No se alarme si no lo entiende todo. **tcpdump** es una herramienta para los profesionales ingenieros de redes y administradores de sistemas. De forma predeterminada, sondea la primera tarjeta de red que encuentre, pero si tiene múltiples interfaces, simplemente use el argumento **-i** para especificar en cuál está interesado. También puede limitar los datos mostrados usando expresiones y

cambiar la manera en que se muestran, pero eso está mejor explicado en la página de manual y otros materiales de referencia.

```
darkstar:~# tcpdump -i wlan0
tcpdump: verbose output suppressed, use -v or -vv for full protocol
decode
listening on wlan0, link-type EN10MB (Ethernet), capture size 96 bytes
13:22:28.221985 IP gw.ctsmacon.com.microsoft-ds > 192.168.1.198.59387:
Flags [P.], ack 838190560, win 3079, options [nop,nop,TS val 1382697489
ecr 339048583], length 164WARNING: Short packet. Try increasing the
snap length by 140
SMB PACKET: SMBtrans2 (REPLY)

13:22:28.222392 IP 192.168.1.198.59387 > gw.ctsmacon.com.microsoft-ds:
Flags [P.], ack 164, win 775, options [nop,nop,TS val 339048667 ecr
1382697489], length 134WARNING: Short packet. Try increasing the snap
length by 110
SMB PACKET: SMBtrans2 (REQUEST)
```

nmap

Suponga que necesita saber ¿qué servicios de red se están ejecutando en una máquina, o múltiples máquinas, o que desea determinar si múltiples las máquinas son receptivas? Podría hacer **ping** a cada una individualmente, hacer **telnet** a cada puerto que le interesara, y anotar cada detalle, pero eso es muy tedioso y lleva mucho tiempo. Una alternativa mucho más fácil es utilizar un escáner de puertos, y **nmap**(1) es justo la herramienta para la tarea.

nmap es capaz de escanear los puertos TCP y UDP, determinar el sistema operativo de un dispositivo de red, probar cada servicio localizado para determinar su tipo específico, y mucho mucho más. Quizás la forma más sencilla de usar **nmap** es hacer “ping” a varias computadoras a la vez. Puede utilizar la notación de dirección de red (CIDR) o especificar un rango de direcciones y **nmap** escaneará cada uno de ellos y devolverá los resultados cuando haya terminado. Incluso puede especificar nombres de host si gusta.

Para “hacer ping” a los hosts, tendrás que usar el argumento **-sP**. El siguiente comando le indica a **nmap** que haga “ping” a www.slackware.com y que las 16 direcciones IP comiencen en 72.168.24.0 y terminen en 72.168.24.15.

```
darkstar:~# nmap -sP www.slackware.com 72.168.24.0/15
```

Si necesita realizar un análisis de puertos, **nmap** tiene muchas opciones para hacer justamente eso. Cuando se ejecuta sin argumentos, **nmap** realiza un escaneo de puerto TCP estándar en todos los hosts especificados. También hay opciones para hacer **nmap** más o menos agresivo con sus escaneos para obtener resultados más rápidos o esquivar los servicios de detección de intrusos. Para una discusión completa, usted debe referirse a la más bien exhaustiva página de manual. Los siguientes tres comandos ejecutan un escaneo de puertos regular, un escaneo SYN y un escaneo “Christmas tree”.

```
darkstar:~# nmap www.example.com
```

```
darkstar:~# nmap -sS www.example.com
darkstar:~# nmap -sX www.example.com
```



¡Esté advertido! Algunos proveedores de servicios de Internet desaprueban en gran medida el escaneo de puertos y pueden tomar medidas para evitar que lo haga. **nmap** y aplicaciones como esta es mejor utilizarlas en sus propios sistemas con fines de mantenimiento y seguridad, no como escáneres de Internet de uso general.

host

A menudo, los problemas de red se derivan de un fallo del DNS (Servicio de Nombres de Dominio) que asigna los nombres de dominio a las direcciones IP. Una manera fácil de realizar una rápida revisión de DNS es el comando **host**(1). Cuando se ejecuta, su ordenador realizará algunas búsquedas de DNS comunes y devolverá los resultados.

```
darkstar:~# host www.slackware.com
www.slackware.com is an alias for slackware.com.
slackware.com has address 64.57.102.34
slackware.com mail is handled by 1 mail-mx.cwo.com.
```

dig

Las búsquedas de DNS más complejas se pueden realizar manualmente con la herramienta **dig**(1). **dig** es “el perro más malo de la perrera” cuando se trata de resolver problemas de DNS. Con esta herramienta, puede realizar prácticamente cualquier búsqueda de DNS desde búsquedas inversas en registros A, CNAME, MX, SP, TXT y más. Hay demasiadas opciones de línea de comandos y tipos de búsqueda para profundizar en ellas aquí, pero la página de manual enumera todos los casos de uso común.

```
darkstar:~# dig @207.69.188.185 www.slackware.com a

; <<>> DiG 9.4.3-P4 <<>> @207.69.188.185 www.slackware.com a
; (1 server found)
;; global options: printcmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 57965
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 0

;; QUESTION SECTION:
;www.slackware.com.      IN      A

;; ANSWER SECTION:
www.slackware.com.      86400   IN      CNAME    slackware.com.
slackware.com.          86400   IN      A        64.57.102.34

;; AUTHORITY SECTION:
```

```
slackware.com.      86400      IN       NS       ns2.cwo.com.
slackware.com.      86400      IN       NS       ns1.cwo.com.

;; Query time: 348 msec
;; SERVER: 207.69.188.185#53(207.69.188.185)
;; WHEN: Sat Jul 3 16:25:10 2010
;; MSG SIZE rcvd: 105
```

Veamos las opciones de la línea de comandos usadas arriba. El argumento `@207.69.188.185` le dice a **dig** qué servidor DNS consultar. Si no se especifica, **dig** simplemente usará los que estén listados en `/etc/resolv.conf`. El argumento `a` al final es el tipo de registro DNS a buscar. En este caso buscamos un registro “A” que ha devuelto una dirección IPv4.

finger

finger(1) no es tanto una herramienta de diagnóstico de red como una herramienta de diagnóstico para el usuario de la red. Usando **finger**(1) puedes juntar una pequeña cantidad de información útil sobre los usuarios en los servidores que ejecutan el demonio **fingerd**(8). Hoy en día muy pocos servidores todavía ofrecen **fingerd**, pero para los que lo hacen puede ser una herramienta útil para hacer un seguimiento de tus amigos y compañeros de trabajo.

```
darkstar:~# finger alan@cardinal.lizella.net
[cardinal.lizella.net]
Login: alan                      Name: Alan Hicks
Directory: /home/alan           Shell: /bin/bash
Office: 478 808 9919, 478 935 8133
On since Wed Apr 13 17:43 (UTC) on pts/9 from
75-150-12-113-atlanta.hfc.comcastbusiness.net
    32 minutes 24 seconds idle
    (messages off)
On since Wed Apr 13 17:45 (UTC) on pts/10 from :pts/9:S.0
    48 minutes 56 seconds idle
Mail forwarded to alan@lizella.net
No mail.
No Plan.
```

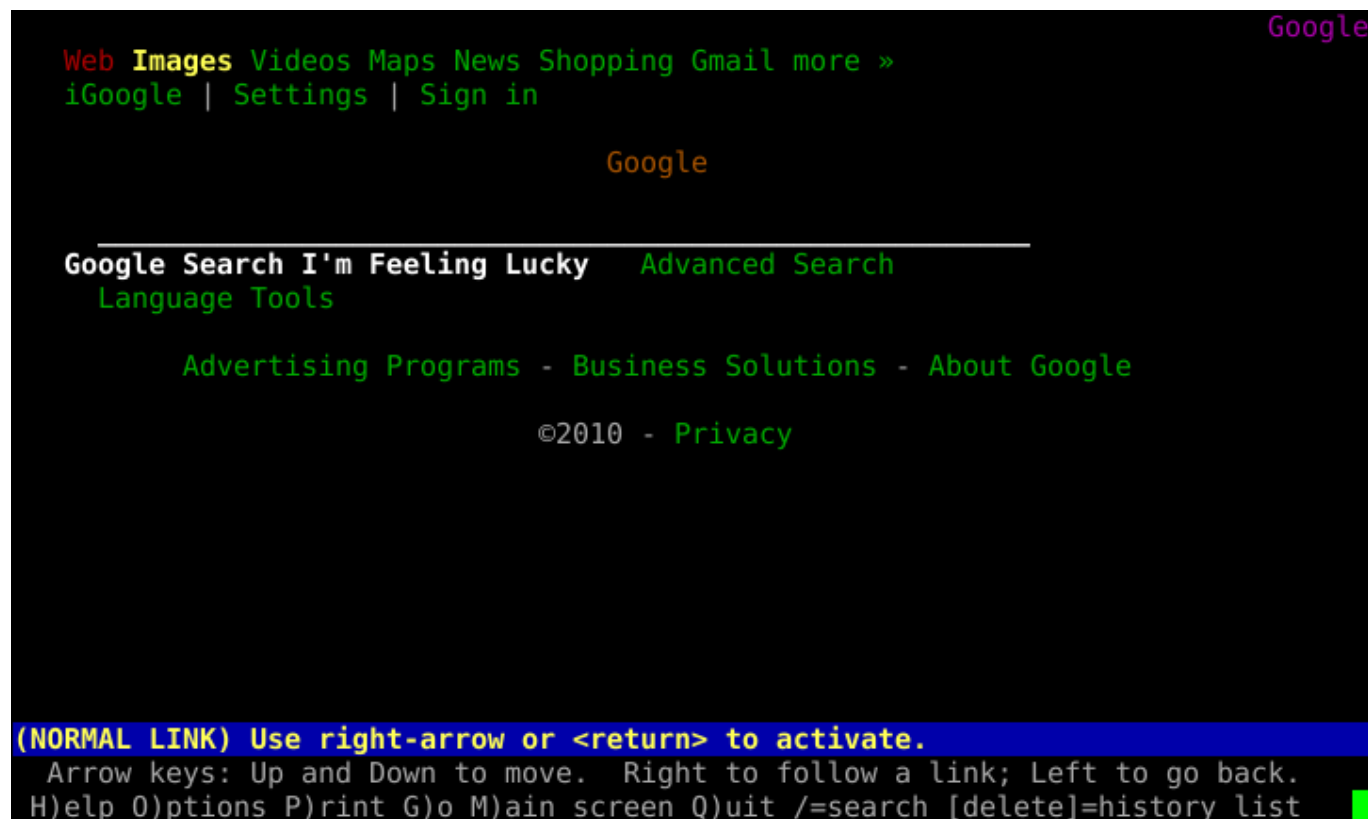
Navegadores Web

Slackware incluye una variedad de navegadores web. Si está usando un entorno de escritorio gráfico, encontrará **Firefox**, **Seamonkey**, y otros con los que esté ya familiarizado, pero ¿qué pasa con el acceso mediante consola? Afortunadamente hay un número de navegadores web capaces aquí también.

lynx

El navegador web basado en consola más antiguo incluido con Slackware es definitivamente **lynx**(1),

un navegador web muy capaz si bien algo limitado. **lynx** no soporta marcos (frames), javascript, o imágenes; es estrictamente un navegador web de texto. La navegación se realiza con las teclas de flecha de su teclado y opcionalmente, un ratón. Si bien carece de muchas características de otros navegadores **lynx** es una de los navegadores web más rápidos. que utilizará para recopilar información. Por ejemplo, el argumento `-dump` envía la página web formateada directamente a la consola que luego puede ser canalizada a otros programas.



```
Web Images Videos Maps News Shopping Gmail more »
iGoogle | Settings | Sign in

Google

Google Search I'm Feeling Lucky Advanced Search
Language Tools

Advertising Programs - Business Solutions - About Google

©2010 - Privacy

(NORMAL LINK) Use right-arrow or <return> to activate.
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

links

Una alternativa más rica en prestaciones es el popular **links**(1), un navegador web basado en consola que soporta marcos y tiene un mejor renderizado de tablas que **lynx**. Como su predecesor, **links** es navegable con las teclas de flechas, y está soportado el uso del ratón. **lynx** también incluye un menú práctico (simplemente haga clic en la línea superior con el ratón para activarlo) y, en general formatea mejor las páginas web.

The Slackware Linux Project

News

Security Advisories

FAQ

Book

General Info

Get Slack

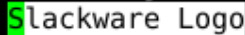
Install Help

Configuration

Packages

ChangeLogs

The Slackware Linux Project (p1 of 15)

 Slackware Logo

Slackware 13.0 is released!

After one of the most intensive periods of development in Slackware's history, the long awaited stable release of Slackware 13.0 is ready. This release brings with it many major changes since Slackware 12.2, including a completely reworked collection of X packages (a configuration file for X is no longer needed in most cases), major upgrades to the desktop environments (KDE version 4.2.4 and Xfce version 4.6.1), a new .txz package format with much better compression, and other upgrades all around -- to the development system, network services, libraries, and major applications like Firefox and Thunderbird. We think you'll agree

2009-08-27

<http://www.slackware.com/index.html>

wget

A diferencia de los otros navegadores que hemos visto, **wget**(1) no es interactivo. En lugar de mostrar el contenido HTTP, **wget** lo descarga. Esto significa el “navegar” fuera del navegador web. A diferencia de los modos de descarga de otros **wget** no formatea el archivo que descarga; más bien copia el contenido en su forma exacta en la web con todas las etiquetas y datos binarios en su lugar. También soporta varias opciones recursivas que pueden reflejar de manera efectiva el contenido en línea en su computadora local. **wget** no necesita funcionar exclusivamente con contenido HTTP; también soporta FTP y varios otros protocolos.

```
darkstar:~# wget
ftp://ftp.osuosl.org/pub/slackware/slackware-current/ChangeLog.txt
--2010-05-01 13:51:19--
ftp://ftp.osuosl.org/pub/slackware/slackware-current/ChangeLog.txt
=> `ChangeLog.txt'
Resolving ftp.osuosl.org... 64.50.236.52
Connecting to ftp.osuosl.org[64.50.236.52]:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.      ==> PWD ... done.
==> TYPE I ... done.    ==> CWD /pub/slackware/slackware-current ... done.
==> SIZE ChangeLog.txt ... 75306
==> PASV ... done.      ==> RETR ChangeLog.txt ... done.
Length: 75306 (74K)

100%[=====>] 75,306          110K/s   in 0.7s

2010-05-01 13:51:22 (110 KB/s) - `ChangeLog.txt' saved [75306]
```


Mail Clients

Slackware también incluye una variedad de clientes de correo electrónico. Si está utilizando un entorno gráfico, encontrará **Thunderbird**, **Kmail**, **sylpheed** y otros. Al igual que con los navegadores web, también hay aplicaciones que funcionan en el shell. Una vez que empiece a utilizar un cliente de correo electrónico en la consola, es posible que no quiera usar nada más; la flexibilidad y la configurabilidad pueden ser adictivas.

pine

pine es uno de los interfaces de clientes de correo en línea de comandos más antiguas todavía en existencia y sigue siendo uno de los más fáciles de usar. **pine** fue creado por la Universidad de Washington y lleva consigo tanto una marca registrada como un licencia de derechos de autor con los que son difíciles de trabajar. Afortunadamente, de vuelta en 2005, la universidad consideró conveniente reescribirlo sin la marca registrada, y con una licencia más abierta, así que **alpine**(1), el clon de pine distribuido con Slackware, nació.

Para empezar a utilizar **alpine**, simplemente escriba **pine** en la línea de comandos. Su uso es muy sencillo gracias a su sistema de menús, así como la referencia de comandos que se encuentra en la parte inferior de la pantalla. Veamos por sí mismo:

```
ALPINE 2.02(1266) MAIN MENU      Folder: INBOX   No Messages

?  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

      Copyright 2006-2008 University of Washington
      Copyright 2009-2010 Re-Alpine Project
      [Folder "INBOX" opened with 0 messages]

? HELP      P PrevCmd      R RelNotes
O OTHER CMDS > [ListFldrs] N NextCmd    K KBlock
```

Antes de configurar cualquier cliente de correo, debe comprobar la documentación de su servidor de correo para reunir toda la información pertinente sobre los protocolos y medidas de seguridad que usa su servicio de correo electrónico. Esto le ayudará a configurar **pine** correctamente. De manera predeterminada, **pine** comprobará si hay nuevos correos electrónicos entregados a un servicio de correo que se ejecute en su ordenador. A menos que esté realmente ejecutando un servicio de correo de este tipo (mucha gente lo hace) esto probablemente no es lo que quiera. Afortunadamente configurar **pine** es un proceso sencillo. Simplemente entre en el menú [S]etup y seleccione la opción [C]onfig. Se le dará la opción de introducir su nombre, ruta de correo, servidor SMTP y muchas otras opciones.

mutt

A algunas personas no les gusta **pine**. Algunas personas quieren más control. Algunas personas quieren un cliente de correo totalmente configurable con soporte para plugins y una actitud sensata. Esa gente usa **mutt**(1). **mutt** no es tan fácil de usar como **pine**, pero lo compensa con potencia. No encontrará la referencia a los comandos en la parte inferior de la pantalla, **mutt** utiliza cada pulgada de su espacio para el servicio de procesamiento de correo. Sus funciones soportadas son extensas ;las pantallas consecutivas no son un problema para el potente mestizo! Puede configurar **mutt** con el archivo “.muttrc” en su directorio de inicio. Con las muchas opciones de configuración posibles, incluso hay una página de manual para ello, muttrc(5). Quizá quiera leer más sobre ello.

```
g:Quit d:Del u:Undel s:Save m:Mail r:Reply g:Group ?:Help
5424 Aug 19 slakmagik (3.9K) Re: [Slackbuilds-users] rc2 slac
5425 Aug 19 LukenShiro (3.0K) Re: [Slackbuilds-users] eric ide
5426 Aug 19 B Watson (3.8K) Re: [Slackbuilds-users] eric ide
5427 Aug 19 Marcin Herda (7.2K) Re: [Slackbuilds-users] i3 updat
5429 Aug 19 Binh Nguyen ( 26K) [Slackbuilds-users] My SlackBuil
5430 Aug 20 rudsonalves ( 77K) Re: [Slackbuilds-users] eric ide
5432 Aug 21 Nicolas Kovacs (3.3K) Re: [Slackbuilds-users] murrine-
5433 N Aug 21 Mr. B-o-B (3.3K) [Slackbuilds-users] unixODBC bui
5434 N Aug 21 Mikko Varri (2.9K) [Slackbuilds-users] REQUIRES inf
5435 N Aug 21 xgizzmo@slackbu (3.3K) Re: [Slackbuilds-users] REQUIRES
5436 N Aug 21 Mikko Varri (4.0K) Re: [Slackbuilds-users] REQUIRES
5437 N Aug 22 Christoph Willi (4.2K) Re: [Slackbuilds-users] REQUIRES
5438 N Aug 21 Robby Workman (5.2K) Re: [Slackbuilds-users] REQUIRES
5439 N Aug 21 King Beowulf (5.6K) [Slackbuilds-users] 13.37 nvidi

-Mutt: =slackbuilds-users [Msgs:5442 New:4 29M]--(date/date)--(100%)--
```

El uso de **mutt** es único porque es por propia naturaleza un Agente de Usuario de Correo (MUA), lo que significa que su verdadero propósito es leer y ordenar el correo electrónico. Este fue su única función originalmente, aunque algunas características adicionales como la recuperación de correo a través de POP3 e incluso una muy básica transferencia a través de SMTP se han colado en la aplicación.

Como suele ocurrir con las aplicaciones robustas basadas en la consola, las opciones de configuración son innumerables, y no hay una manera “correcta” o “errónea” de usar **mutt** siempre y cuando haga lo que quiera que haga. Una cosa a tener en cuenta si usted está considerando usar **mutt** para el correo es que su capacidad de envío y recepción de correo es muy limitada. **mutt** se centra únicamente en la clasificación, lectura y redacción de mensajes de correo, además de las tradicionales funciones de agente de correo de usuario. Esto mantiene el foco con la filosofía de UNIX de pequeñas herramientas que hacen una cosa muy bien y que pueden combinarse (o “encadenarse”) con otras herramientas para completar cualquier tarea necesaria. Con esto en mente, es probable que necesite configurar alguna herramienta externa para recibir correo, como mínimo.

Los comandos utilizados para navegar por **mutt** son altamente configurables pero los predeterminados se pueden listar tecleando `?`.

mailx

Así que son geniales y todo eso, ¿pero qué pasa si sólo quieres un correo que no está basado en menús? Afortunadamente **mailx** está aquí para salvarte.

mailx está basado en la aplicación Berkeley Mail con un comando **mail** que aparece en la Versión 1 del UNIX de AT&T. Puede ser utilizados de forma interactiva o no interactiva.

mailx lee el correo de la computadora y muestra la combinación habitual de remitente, asunto, estatus, y el tamaño en una lista, dejando al usuario en una línea de consola interactiva. De hecho, podría parecerle familiar si se tomó la molestia de revisar su correo inmediatamente después de instalar Slackware y leyó el saludo de Pat Volkerding.

```
darkstar:~# mailx
Heirloom mailx version 12.4 7/29/08.Type ? for help.
"/var/spool/mail/root": 2 messages 2 new
>N1 To rootThu Mar 10 23:33 52/1902Register with the Linux counter project
  N2 To rootThu Mar 10 23:35321/15417 Welcome to Linux (Slackware 14.0)!
?;
```

Para leer un mensaje, introduzca el número del mensaje en el campo en el momento oportuno. Esto muestra el mensaje utilizando **more** así que utilice la tecla **RETORNO** para ver la página siguiente. Una vez que ha alcanzado el final del mensaje pulse **q** para volver a la vista de lista, o bien **RETORNO** para continuar con el siguiente mensaje.

Para ver una lista de los comandos disponibles, introduzca **?** en el campo **mail**; usando los comandos proporcionados, puede visualizar las cabeceras de correo en el SPOOL, responder, borrar, grabar, y muchas otras tareas comunes de correo electrónico.

mailx es más potente cuando se usa en scripts. Para todas las opciones disponibles para **mailx**, vea su página de manual. Una manera sencilla de enviar un correo electrónico a alguien requiere sólo el comando en sí y la dirección de destino.

```
darkstar:~$ mailx bob@example.com
```

Después de que se haya ejecutado el comando, aparece una línea de comandos interactiva para la línea de asunto, el cuerpo del mensaje y el carácter final (un simple punto en una línea vacía).

mailx puede ser completamente utilizado sin intervención humana, sin embargo. Generalmente, es seguro asumir que cualquier atributo que se pueda definir en el shell interactivo para **mailx** también se puede definir durante la creación de scripts o usándolo como un comando no interactivo.

```
darkstar:~$ mailx -n -s "Test message" bob@example.com < ~/message.txt
```

En este ejemplo, el contenido del archivo "message.txt" sería enviado como el cuerpo del mensaje al destinatario especificado. No se requiere la interacción del usuario.

Dentro de su propio ordenador (localhost) o de su propia red, enviar correo de esta manera es perfectamente posible. Pero a través de Internet generalmente se requieren algunos pasos más. Por supuesto, normalmente hay un servidor smtp que se encarga de la entrega de su correo electrónico. Esto también se puede especificar como parte de su comando **mail**:

```
darkstar:~$env MAILRC=/dev/null
from="bob@example.com (Bob Dobbs)"
smtp=relay.example.com mail -n -s "Test message" connie@example.com <
~/message.txt
```

En este caso, la variable *MAILRC* se fija en */dev/null* para poder anular cualquier valor por defecto del sistema, y el servidor *smtp* así como la línea *FROM*: se definen. El resto del comando es el mismo que usar **mailx** internamente dentro de la propia computadora o red.

Sobre todo, **mailx** es usualmente visto como un cliente de correo con las prestaciones mínimas; esto es en gran medida cierto, pero cuando necesita ser capaz de hacer un script enviando correos electrónicos de notificación o mensajes de actualización importantes, rápidamente se vuelve mucho más valioso que una aplicación completamente interactiva como o **pine** o **mutt**.

Cientes FTP

Muchos datos se almacenan en servidores FTP en todo el mundo. De hecho, Slackware Linux se ofreció por primera vez públicamente a través de FTP y sigue siendo distribuido de esta manera hoy. La mayoría del software de código abierto puede ser descargado en código fuente o en forma binaria a través de FTP, por lo que saber cómo recuperar esta información es una habilidad muy útil.

ftp

El cliente FTP más simple incluido con Slackware se llama simplemente, **ftp**(1) y es un fiable, aunque algo simple medio de envío y recuperación de datos. **ftp** se conecta a un servidor FTP, le pide su nombre de usuario y contraseña, y entonces le permite poner o recibir datos desde y hacia ese servidor. **ftp** ha caído en desgracia con los usuarios más experimentados ante la falta de funcionalidades, pero sigue siendo una herramienta útil, y gran parte de la documentación que ve en línea le remitirá a ella.

Una vez que se ha inicializado una sesión FTP, se le colocará en una línea de comandos similar a una de shell. Desde aquí puede cambiar y listar directorios usando los comandos “*cd*” y “*ls*”, igual que en una shell. Adicionalmente, puede ejecutar el comando “*put*” para enviar un archivo al servidor, o un “*get*” para recuperar datos del servidor. Si se está conectando a un servidor FTP público, querrá usar el nombre de usuario “*anónimo*” y simplemente introduzca su dirección de correo electrónico (o una falsa) para obtener la contraseña.

```
darkstar:~$ ftp ftp.osuosl.org
Name (ftp.osuosl.org:alan): anonymous
331 Please specify the password.
Password: secret
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd pub/slackware/slackware-current/
250 Directory successfully changed.
ftp> get ChangeLog.txt
```

```
local: ChangeLog.txt remote: ChangeLog.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for ChangeLog.txt (33967
bytes).
226 File send OK.
33967 bytes received in 0.351 secs (94 Kbytes/sec)
ftp> bye
221 Goodbye.
```

ncftp

ncftp(1) (se pronuncia nick-f-t-p), es un sucesor más rico en prestaciones que **ftp**, soportando compleción de tabulador y recuperación recursiva. Se conecta automáticamente a un servidor como usuario anónimo, a menos que especifique un nombre de usuario diferente en la línea de comandos con el argumento **-u**. La principal ventaja sobre **ftp** es la capacidad de enviar y recuperar varios archivos a la vez con los comandos “**mput**” y “**mget**”. Si le pasas el argumento **-R** a cualquiera de ellos, pondrán u obtendrán datos recursivamente de los directorios.

```
darkstar:~# ncftp ftp.osuosl.org
Logging in...
Login successful.
Logged in to ftp.osuosl.org.
ncftp / > cd pub/slackware/slackware-current
Directory successfully changed.
ncftp ...ware/slackware-current > mget -R isolinux
isolinux/README.TXT: 4.63 kB 16.77 kB/s
isolinux/README_SPLIT.TXT: 788.00 B5.43 kB/s
isolinux/f2.txt: 793.00 B5.68 kB/s
isolinux/initrd.img:13.75 MB 837.91 kB/s
isolinux/iso.sort:50.00 B354.50 B/s
isolinux/isolinux.bin:14.00 kB 33.99 kB/s
isolinux/isolinux.cfg: 487.00 B3.30 kB/s
isolinux/message.txt:760.00 B5.32 kB/s
isolinux/setpkg: 2.76 kB 19.11 kB/s
ncftp ...ware/slackware-current > bye
```

lftp

El último cliente que vamos a ver es **lftp**(1). Como **ncftp**, soporta compleción de tabulador y actividad recursiva, pero tiene una licencia más amigable. Más bien que usando “**mget**” y “**mput**”, todas las operaciones recursivas se manejan con el comando “**mirror**”. “**mirror**” tiene muchas opciones diferentes disponibles, así que tendré que dirigirte a la página de manual y al comando incorporado “**help**” para más detalles.

```
darkstar:~# lftp ftp.osuosl.org
lftp ftp.osuosl.org:~> cd /pub/slackware/slackware-current
cd ok, cwd=/pub/slackware/slackware-current
lftp ftp.osuosl.org:/pub/slackware/slackware-current> mirror isolinux
```

```
Total: 2 directories, 16 files, 1 symlink
New: 16 files, 1 symlink
14636789 bytes transferred in 20 seconds (703.7K/s)
lftp ftp.osuosl.org:/pub/slackware/slackware-current> bye
```

rsync

¿Listo para ver algo genial? ¿Alguna vez se ha encontrado a sí mismo necesitando sólo un puñado de archivos de un directorio grande, pero no está del todo seguro de qué archivos ya dispone y cuáles necesita? Puede descargar el directorio entero de nuevo, pero eso es duplicar mucho trabajo. Usted puede escoger y elegir, revisar manualmente todo, pero eso es muy tedioso. ¿Tal vez ha descargado un archivo grande como un ISO, pero algo salió mal con la descarga? No tiene sentido que usted deba volver a descargar el archivo completo si sólo se han corrompido unos pocos bits. Introduzca **rsync**(1), una rápida y versátil herramienta de copia para archivos locales y remotos.

rsync usa un grupo de simples, pero muy efectivas técnicas para determinar qué es lo que hay que cambiar. Verificando el tamaño de archivo y la marca de tiempo, puede determinar si dos archivos son diferentes. Si algo ha cambiado, puede determinar qué bytes son diferentes, y simplemente descargar ese puñado de datos en lugar de un archivo entero. Es realmente una maravilla de la tecnología moderna.

En su forma simplista, **rsync** se conecta a un protocolo rsync y descarga una lista de archivos y directorios, junto con sus tamaños, marcas de tiempo y otra información. A continuación, la compara con los archivos locales (si los hay) para determinar lo que necesita para la transferencia. Sólo se sincronizarán los archivos que sean diferentes. Adicionalmente divide los archivos grandes en trozos más pequeños y compara esos trozos utilizando una función de hash rápida y sencilla. Cualquier trozo que coincida no será transferido, de modo que la cantidad de datos que deben copiarse puede ser dramáticamente reducida. **rsync** también soporta compresión, verbosidad, borrado de archivos, manejo de permisos y muchas otras opciones. Para obtener una lista completa, deberá consultar la sección de manual pero he incluido una pequeña tabla con algunas de las más comunes.

Table 16.1. rsync Arguments

-v	Verbosidad incrementada
-c	Compruebe la suma de todos los archivos en lugar de confiar en el tamaño del archivo y la fecha y hora.
-a	Modo archivo (equivalente a -rlptgoD)
-e	Especifica una shell remota a usar
-r	Modo recursivo
-u	Actualizar - omitir archivos que son más nuevos en el extremo receptor
-p	Preservar permisos
-n	Ejecutar en prueba - realizar una ejecución de prueba sin hacer ningún cambio
-z	Compresión - práctico para conexiones de red lentas

Debido a la potencia y versatilidad de **rsync**, se puede invocar de varias maneras. Los dos ejemplos siguientes se conectan a un servidor de protocolo rsync para recuperar alguna información y a otro servidor vía ssh para encriptar la transmisión.

```
darkstar:~# rsync -avz rsync://ftp.osuosl.org/pub/slackware/slackware-  
current/ \  
/src/slackware-current/  
darkstar:~# rsync -e ssh ftp.slackware.com:/home/alan/foo /tmp/foo
```

Navegación de Capítulos

Capítulo Anterior: [Redes inalámbricas](#)

Capítulo siguiente: [Gestión de paquetes](#)

Fuentes

- Fuente original: <http://www.slackbook.org/beta>
- Escrito originalmente por Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson
- Traducido por [Pedro Herrero García](#) 2019/02/05 19:41 (UTC)

[slackbook](#), [networking](#), [mail clients](#), [web browsers](#), [rsync](#)

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
https://docs.slackware.com/es:slackbook:basic_networking_utilities

Last update: **2019/03/03 17:07 (UTC)**

