Basic Networking Utilities

So you've finally managed to setup your network connection, now what? How do you know that it's working? How do you know that you set it up correctly? And just what do you do now that it's setup? Well this chapter is for you.

Network Diagnostic Tools

Slackware Linux includes a great many networking tools for troubleshooting and diagnosing network connection troubles, or just for seeing what's out there on the network. Most of these tools are command-line tools, so you can run them from a virtual terminal or in a console window on your graphical desktop. A few of them even have graphical front-ends, but we're going to deal almost exclusively with command-line tools for now.

ping

ping(8) is a handy tool for determining if a computer is operational on your network or on the Internet at large. You can think of as a type of sonar for computers. By using it, you send out a *"ping"* and listen for an echo to determine if another computer or network device is listening. By default, ***ping*** checks for the remote computer once per second indefinitely, but you can change the interval between checks and the total number of checks easily, just check the man page. You can terminate the application at any time with CTRL+c. When ***ping*** is finished, it displays a handy summary of its activity. ***ping*** is very useful for determining if a computer on your network or the Internet is available, but some systems block the packets ***ping*** sends, so sometimes a system may be functioning properly, but still not send replies.

```
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) is a handy tool for determining what route your packets take to reach some other computer. It's mainly of use for determining which computers are *"near"* or *"far"* from you. This distance isn't strictly geographical, as your Internet Service Provider may route traffic from your computer in strange ways. ***traceroute*** shows you each router between your computer and any other machine you wish to connect to. Unfortunately, many providers, firewalls, and routers will block ***traceroute*** so you might not get a complete picture when using it. Still, it remains a handy tool for network troubleshooting.

```
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

Once upon a time, ***telnet***(1) was the greatest thing since sliced bread. Basically, ***telnet*** opens an unencrypted network connection between two computers and hands control of the session to the user rather than some other application. Using ***telnet***, people could connect to shells on other computers and execute commands as if they were physically present. Due to its unencrypted nature this is no longer recommended; however, ***telnet*** is still used for this purpose by many devices.

Today, ***telnet*** is put to better use as a network diagnostic tool. Because it passes control of the session directly to the user, it can be used for a great variety of testing purposes. As long as you know what ASCII commands to send to the receiving computer, you can do any number of activies, such as read web pages or check your e-mail. Simply inform ***telnet*** what network port to use, and you're all set.

```
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

As we mentioned, ***telnet*** may be useful as a diagnostic tool, but its unencrypted nature makes it a security concern for shell access. Thankfully, there's the secure shell protocol. Nearly every Linux, UNIX, and BSD distribution today makes use of OpenSSH, or ***ssh***(1) for short. It is one of the most commonly used network tools today and makes use of the strongest cryptographic techniques. ***ssh*** has many features, configuration options, and neat hacks, enough to fill its own book, so we'll only go into the basics here. Simply run ***ssh*** with the user name and the host and you'll be connected to it quickly and safely. If this is the first time you are connecting to this computer, ***ssh*** will ask you to confirm your desire, and make a local copy of the encryption key to use. Should this key later change, ***ssh*** will warn you and refuse to connect because it is possible that some one is attempting to hijack the connection using what is known as a man-in-the-middle attack.

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

The user and hostname are in the same form used by e-mail addresses. If you leave off the username part, ***ssh*** will use your current username when establishing the connection.

tcpdump

So far all the tools we've looked at have focused on making connections to other computers, but now we're going to look at the traffic itself. ***tcpdump***(1) (which must be run as root) allows us to view all or part of the network traffic originating or received by our computer. ***tcpdump*** displays the raw data packets in a variety of ways with all the network headers intact. Don't be alarmed if you don't understand everything it displays, ***tcpdump*** is a tool for professional network engineers and system administrators. By default, it probes the first network card it finds, but if you have multiple interfaces, simply use the *-i* argument to specify which one you're interested in.You can also limit the data displayed using expressions and change the manner in which it is displayed, but that is best explained by the man page and other reference material.

```
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

Suppose you need to know what network services are running on a machine, or multiple machines, or you wish to determine if multiple machines are responsive?You could ***ping*** each one individually, ***telnet*** to each port you're interested in, and note every detail, but that's very tedious and time consuming. A much easier alternative is to use a port scanner, and ***nmap***(1) is just the tool for the job.

nmap is capable of scanning TCP and UDP ports, determining the operating system of a network device, probing each located service to determine its specific type, and much much more.Perhaps the simplist way to use ***nmap*** is to *"ping"* multiple computers at once. You can use network address notation (CIDR) or specify a range of addresses and ***nmap*** will scan every one and return the results to you when it's finished. You can even specify host names as you like.

In order to *"ping"* hosts, you'll have to use the *-sP* argument. The following command instructs ***nmap*** to *"ping"* www.slackware.com and the 16 IP addresses starting at 72.168.24.0 and ending at 72.168.24.15.

```
darkstar:~# nmap -sP www.slackware.com 72.168.24.0/28
```

Should you need to perform a port scan, ***nmap*** has many options for doing just that. When run without any arguments, ***nmap*** performs a standard TCP port scan on all hosts specified. There are also options to make ***nmap*** more or less aggressive with its scanning to return results quicker or fool intrusion detection services. For a full discussion, you should refer to the rather exhaustive man page. The following three commands perform a regular port scan, a SYN scan, and a *"Christmas tree"* scan.

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

Be warned! Some Internet Service Providers frown heavily on port scanning and may take measures to prevent you from doing it. ***nmap*** and applications like it are best used on your own systems for maintenance and security purposes, not as general purpose Internet scanners.

host

Often network problems stem from a failure of DNS (Domain Name Service) which maps domain names to IP addresses. An easy way to perform quick DNS lookups is the ***host***(1) command. When this is run, your computer will perform a few common DNS lookups and return the results.

```
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

More complex DNS lookups can be manually performed with the ***dig***(1) tool.***dig*** is *"the meanest dog in the pound"* when it comes to troubleshooting DNS issues. With this tool, you can perform virtually any DNS lookup from reverse lookups to A, CNAME, MX, SP, TXT records and more. There are far too many command-line options and lookup types to go into depth here, but the man page lists all the common use cases.

```
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
```

Let's take a took at the command-line options used above.The *@207.69.188.185* argument tells ***dig*** what DNS server to query. If it is not specified, ***dig*** will simply use whatever servers are listed in `/etc/resolv.conf`.The *a* argument at the end is the type of DNS record to lookup. In this case we looked for an *"A"* record which returned an IPv4 address.

finger

finger(1) isn't exactly a network diagnostic tool as much as it is a network-user diagnostic tool. Using ***finger***, you can gather a handful of useful information about users on servers running the ***fingerd***(8) daemon. Today very few servers still offer ***fingerd***, but for those that do it can be a useful tool for keeping track of your friends and co-workers.

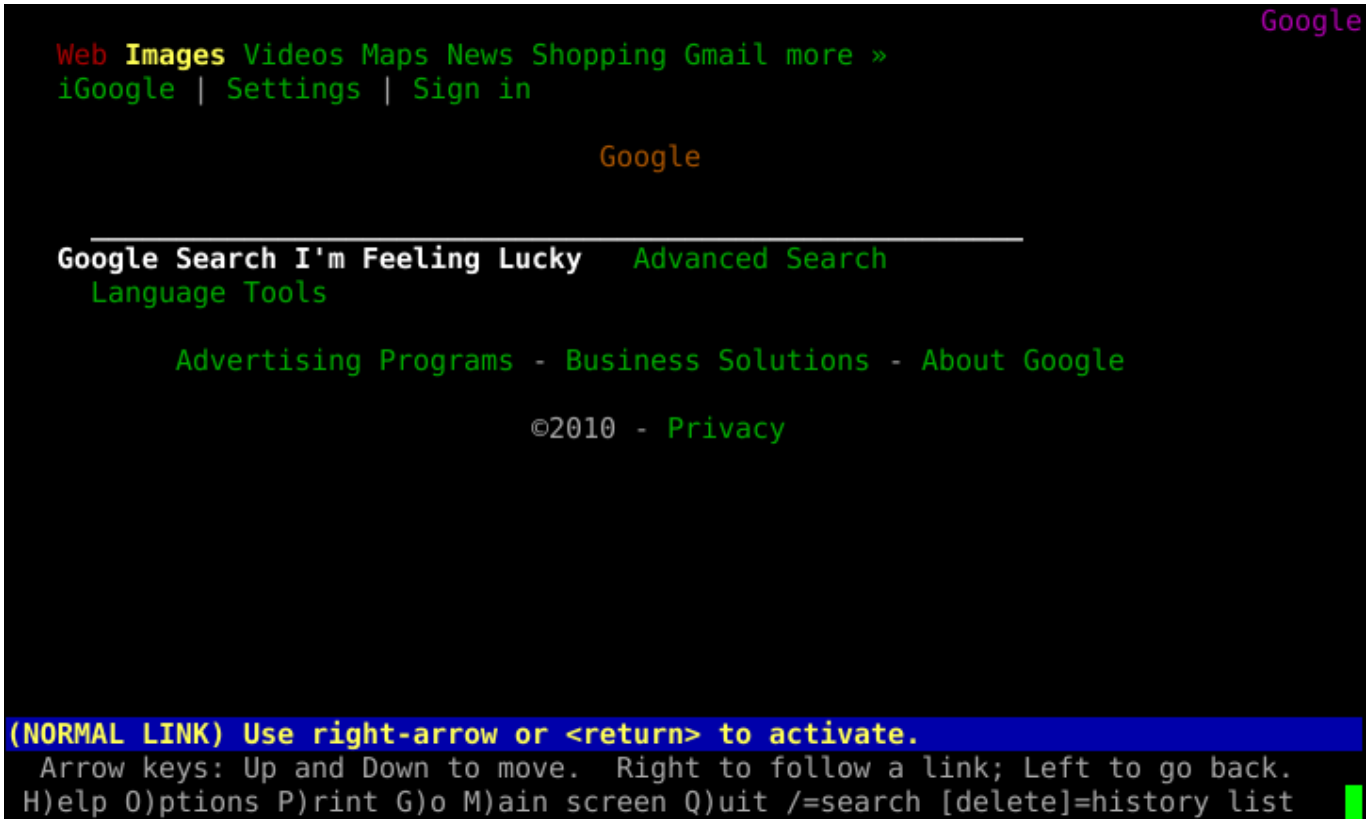
```
darkstar:~# finger alan@cardinal.lizella.net
[cardinal.lizella.net]
Login: alan                     Name: Alan Hicks
Directory: /home/alan                   Shell: /bin/bash
Office: 478 808 9919, 478 935 8133
On since Wed Apr 13 17:43 (UTC) on pts/9 from
75-150-12-113-atlanta.hfc.comcastbusiness.net
    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.
```

Web Browsers

Slackware includes a variety of web browsers. If you're using a graphical desktop, you'll find ***Firefox***, ***Seamonkey***, and others you may already be familiar with, but what about console access? Fortunately, there are a number of capable web browsers here as well.

lynx

The oldest console-based web browser included with Slackware is definitely ***lynx***(1), a very capable if somewhat limited web browser.***lynx*** does not support frames, javascript, or pictures; it is strictly a text web browser. Navigation is performed using your keyboard's arrow keys and optionally, a mouse. While it lacks many features that other browsers support, ***lynx*** is one of the fastest web browsers you'll ever use for gathering information. For example, the *-dump* argument sends the formatted web page directly to the console, which can then be piped to other programs.



links

A more feature-rich alternative is the popular ***links***(1), a console-based web browser that supports frames and has better table rendering than ***lynx***. Like its predecessor, ***links*** is navigable with the arrow keys, and the use of a mouse is supported.Unlike ***lynx***, it also includes a handy menu (simply click on the top line with your mouse to activate) and generally formats web pages better.

wget

Unlike the other browsers we've looked at, ***wget***(1) is non-interactive. Rather than display HTTP content, ***wget*** downloads it. This takes the *"browsing"* out of the web browser. Unlike the dump modes of other browsers, ***wget*** does not format its downloads; rather it copies the content in its exact form on the web server with all tags and binary data in place. It also supports several recursive options that can effectively mirror online content to your local computer.***wget*** need not operate exclusively on HTTP content; it also supports FTP and several other protocols.

```
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]
```

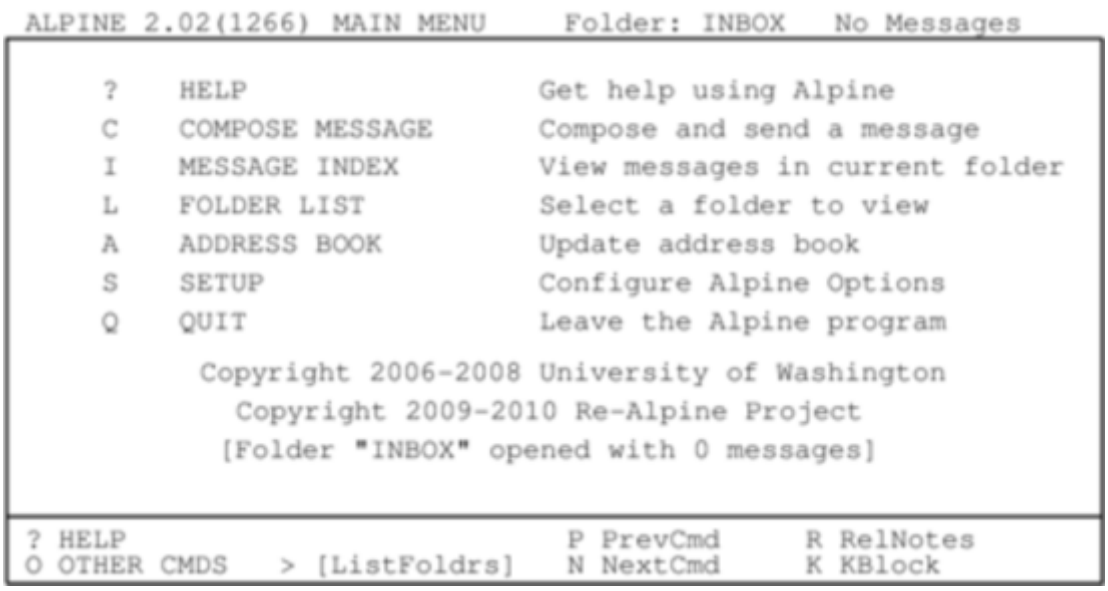
Mail Clients

Slackware also includes a variety of email clients. If you're using a graphical desktop, you'll find ***Thunderbird***, ***Kmail***, ***sylpheed*** and others. As with web browsers, there are also applications that function within the shell. Once you start using an email client in the console, you may find yourself not wanting to use anything else; the flexibility and configurability can be addicting.

pine

pine is one of the oldest command-line interface mail clients still in existance and remains one of the most user-friendly. ***pine*** was created by the University of Washington and carries with it both a trademark and a copyright license that are difficult to work with. Thankfully back in 2005, the university saw fit to re-write it without the trademark and with a more open license, so ***alpine***(1), the pine-clone distributed with Slackware, was born.

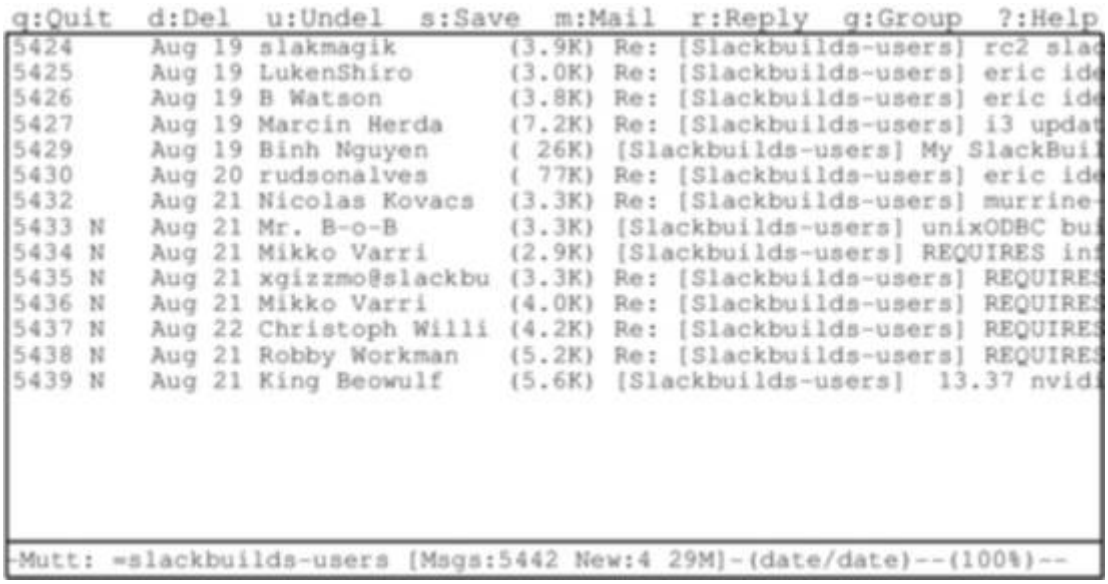
To start using ***alpine***, simply type ***pine*** at the command line. Using it is very simple due to its menu-driven system as well as the command reference neatly located at the bottom of the screen. See for yourself:



Before configuring any mail client, you should check the documentation of your mail server to gather all of the pertinent information about what protocols and security measures your mail service uses. This will help you configure ***pine*** correctly. By default, ***pine*** will check for new e-mails delivered to a mail service running on your computer. Unless you're actually running such a mail service (many people do) this probably isn't what you want. Fortunately configuring ***pine*** is a straight forward process. Simply enter the [S]etup menu and chose the [C]onfig option. You'll be given an option to enter you name, mail path, SMTP server, and many other options.

mutt

Some people don't like ***pine***. Some people want more control. Some people want a fully-configurable mail client with plugin support and a no-nonsense attitude. Those people use ***mutt***(1). ***mutt*** isn't as user friendly as ***pine***, but makes up for it with power. You won't find the user-friendly command reference at the bottom of the screen, ***mutt*** uses every last inch of real-estate for mail processing duty. It's feature support is extensive - threaded displays are no problem for the mighty mixed-breed! You can configure ***mutt*** with a `.muttrc` file in your home directory. With all the many different possible configuration options, there's even a man page for that, muttrc(5). You might want to read up on it.



Using ***mutt*** is unique because it is by nature a Mail User Agent (MUA), meaning its true purpose is to read and sort email. This was its only job originally, although some additional features such as retrieving mail via POP3 and even very basic transfering messages via SMTP have snuck into the application.

As is so often the case with robust console-based applications, the configuration options are myriad, and there is no *"right"* or *"wrong"* way of using ***mutt*** as long as it does what you want it to do. One thing to keep in mind if you are considering using ***mutt*** for mail handling is that its mail sending and receiving abilities are very limited. ***mutt*** focuses solely on sorting, reading, and composing mail messages in addition to other traditional Mail User Agent duties. This is keeping in focus with the UNIX philosophy of small tools that do one thing very well and which can be combined (or *"chained"*) with other tools to complete whatever tasks are required. With this in mind, you'll likely need to setup some external tool to receiving mail at a minimum.

The commands used to navigate around in ***mutt*** are highly customizable but the defaults can be listed by typing ?.

mailx

So those are great and everything, but what if you just want a mail client that isn't menu-driven? Thankfully ***mailx*** is here to save you.

mailx is based on the Berkeley Mail application, with a ***mail*** command appearing as early as Version 1 of AT&T's UNIX. It can be used either interactively or non-interactively.

mailx reads mail from your computer's mail spool and displays the usual combination of sender, subject, status, and size in a list, leaving the user at an interactive prompt. In fact, it might look familiar to you if you bothered checking your mail immediately after installing Slackware and read Pat Volkerding's greeting.

```
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)!
?;
```

To read a message, enter the number of the message at the prompt. This displays the message using ***more***, so use the RETURN key to view the next page. Once the end of the message has been reached, press q to return to the list view, or RETURN to continue to the next message.

To see a list of available commands, enter ? at the ***mail*** prompt; using the commands provided, you can view the headers of mail in the spool, reply, delete, save, and many other common email tasks.

mailx is most powerful when used in scripting. For all of the options available for ***mailx***, view its man page. A simple way to send an email to someone requires only the command itself and the destination address.

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

After the command has been issued, an interactive prompt appears for a subject line, the message body, and the end character (a single period on an otherwise empty line).

mailx can be used entirely without human intervention, however. Generally, it's safe to assume that any attribute you can define in the interactive shell for ***mailx*** can also be defined while scripting it or using it as one non-interactive command.

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

In this example, the contents of the file `message.txt` would be sent as the message body to the specified recipient. No interaction from the user is required.

Within one's own computer (localhost) or one's own network, sending email in this manner is entirely possible. But over the internet a few more steps are usually required along the way. Of course, most notably there is usually an smtp server handling the delivery of your email. This, too, can be specified as part of your ***mail*** command:

```
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
```

In this case, the *MAILRC* variable is set to /dev/null in order to override any system defaults, and the smtp server as well as the *FROM:* line are defined. The rest of the command is the same as using ***mailx*** internally within one's own computer or network.

Over all, ***mailx*** is usually viewed as a mail client with the bare-minimum features; this is largely true, but when you need to be able to script sending notification emails or important update messages, it quickly becomes a lot more valuable than a fully interactive application like ***pine*** or ***mutt***.

FTP Clients

Lots of data is stored on FTP servers the world over. In fact, Slackware Linux was first publically offered via FTP and continues to be distributed in this fashion today. Most open source software can be downloaded in source code or binary form via FTP, so knowing how to retrieve this information is a handy skill.

ftp

The simplest FTP client included with Slackware is named simply, ***ftp***(1) and is a reliable if somewhat simple means of sending and retrieving data. ***ftp*** connects to an FTP server, asks for your username and password, and then allows you to put or get data to and from that server. ***ftp*** has fallen out of favor with more experienced users do to a lack of features, but remains a handy tool, and much of the documentation you see online will refer you to it.

Once an FTP session has been initialized, you'll be placed at a prompt somewhat like a shell.From here you can change and list directories using the *"cd"* and *"ls"* commands, just like a shell.Additionally, you may issue the *"put"* command to send a file to the server, or a *"get"* command to retrieve data from the server. If you're connecting to a public FTP server, you'll want to use the *"anonymous"* username and simply enter your e-mail address (or a fake one) for the password.

```
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) (pronounced nick-f-t-p), is a more feature rich successor to ***ftp***, supporting tab completion and recursive retrieval. It automatically connects to a server as the anonymous user, unless you specify a different username on the commandline with the *-u* argument. The primary advantage over ***ftp*** is the ability to send and retrieve multiple files at once with the *"mput"* and *"mget"* commands. If you pass the *-R* argument to either of them, they will recursively put or get data from directories.

```
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

The last client we're going to look at is ***lftp***(1).Like ***ncftp***, it supports tab completion and recursive activity, but has a more friendly license.Rather than user *"mget"* and *"mput"*, all recursive operations are handled with the *"mirror"* command.*"mirror"* has many different options available, so I'll have to refer you to the man page and the built-in *"help"* command for complete details.

```
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

Ready to see something cool? Have you ever found yourself needing just a handful of files from a large directory, but you're not entirely sure which files you already have and which ones you need? You can download the entire directory again, but that's duplicating a lot of work. You can pick and chose, manually check everything, but that's very tedious. Perhaps you've downloaded a large file such as an ISO, but something went wrong with the download? It doesn't make sense that you should have to pull down the entire file again if only a few bits have been corrupted.Enter ***rsync***(1), a fast and versatile copying tool for local and remote files.

rsync uses a handful of simple, but very effective techniques to determine what needs to be changed. By checking file size and time stamps, it can determine if two files are different. If something has changed, it can determine what bytes are different, and simply download that handfull of data rather than an entire file. It is truly a marvel of modern technology.

In its simplist form, ***rsync*** connects to an rsync protocol server and downloads a list of files and directories, along with their sizes, timestamps, and other information. It then compares this to the local files (if any) to determine what it needs to transfer. Only files that are different will be synced.Additionally, it breaks up large files into smaller chunks and compares those chunks using a quick and simple hash function. Any chunks that match are not transferred, so the amount of data that must be copied can be dramatically reduced.***rsync*** also supports compression, verbose output, file deletion, permission handling, and many other options. For a complete list, you'll need to refer to the man page, but I've included a small table of some of the more common options.

Table 16.1. rsync Arguments

-v	Increased verbosity
-c	Checksum all files rather than relying on file size and timestamp
-a	Archive mode (equivilant to -rlptgoD)
-e	Specify a remote shell to use

-r	Recursive mode
-u	Update - skip files that are newer on the receiving end
-p	Preserve permissions
-n	Dry-run - perform a trial run without making any changes
-z	Compress - handy for slow network connections

Due to the power and versatility of ***rsync***, it can be invoked in a number of ways.The following two examples connect to an rsync protocol server to retrieve some information and to another server via ssh to encrypt the transmission.

```
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
```

Chapter Navigation

Previous Chapter: Wireless Networking

Next Chapter: Package Management

Sources

- Original source: http://www.slackbook.org/beta
- Originally written by Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson

slackbook, networking, mail clients, web browsers, rsync