UnixWorld ‘‘New to Unix’’ Column: October 1988

Introducing Unix Mail

By Augie Hansen

[*Editor’s note*: We’ll surround text to be typed with actual double quotes, like: enter: "vi file1". Of course, don’t type the quotes unless so instructed. We’ll use ‘‘balanced quotes’’, constructed using pairs of backward and forward quotes, for all other purposes.]

Unix mail is an example of electronic mail, which is usually referred to as e-mail. E-mail is becoming an increasingly important way to exchange information as computers are accepted and integrated into the workplace.

Because computers have not yet achieved the ubiquitous status of telephones and mailboxes, e-mail is not a universal solution to your communications problems, but it can be a real benefit in many office and personal settings.

Avoiding ‘‘telephone tag’’ is one good use of e-mail. We can view telephone calls as *synchronous* communications--both participants must be available and willing to communicate at the same time. In a typical office setting, this requirement is met less than 30 percent of the time. E-mail is a form of *asynchronous* communication, which permits individuals to communicate when time is available.

Although the elapsed time for a complete e-mail communication may seem long compared to an equivalent telephone call, it really is more efficient. The benefits of e-mail are numerous. Here are a few:

- You don’t waste time making repeated attempts to get through on the telephone.
- You don’t interrupt someone else’s meetings or work sessions.
- You can send a message at any time of the day and night.
- You can call in from anywhere to read your mail or send mail to other users if your system has remote access by phone.
- You can call in from anywhere to read your mail or send mail to other users if your system has remote access by phone.
- You can communicate with users on other systems if your system is tied in to a network.

The primary disadvantage of e-mail is that all users must make a conscious effort to check for received mail. If they are logged in full time, the UNIX system can alert them of the arrival of new mail. If they are not operating full time, they must log in periodically to check for mail.

Sending and Receiving Mail

[*Editor's note:* the following examples illustrate the older ‘‘bin’’ mail program. The newer Berkeley (capital-M) Mail and System V mailx programs will behave differently. However, the concepts are valid for any mail reading program, known today as a Mail User Agent or MUA.--*Becca Thomas.*]

I'll illustrate the use of e-mail by using several examples derived from my daily work. Listing 1A shows a sample e-mail message sent to me by an associate. The fact that some mail is waiting for me is announced when I log in. To read the mail. I type the mail command and carry on a mildly interactive session.

After displaying a message on-screen, the MUA prompts me by showing a question mark on-screen and waits for a command. Mail recognizes single-character commands. By pressing "d", I tell the mail reader to delete the message. Because it was the only message waiting to be read, the program quits and returns control to the shell. Had there been other messages, they would have been displayed one at a time.

If you forget the mail program commands or haven't learned them yet, you can type the "?" command to display a command summary (see Listing 1B).

You can delete or save mail messages, send mail to other users, and move forward and backward through your received mail by using these commands. For example, to save a mail item, type "s" followed by by a file name that you create. Separate the command from the file name with a space or tab. This command appends the current message to the named file (or creates the file if it doesn't exist). If you don't provide a file name, the program uses the name ‘‘mbox’’ in your home (log-in) directory. You can send mail to another user on your system by invoking your mail program with an argument that is the log-in name of the recipient. Listing 1C shows my reply to the previously received mail item. I used the log-in name ‘‘dab’’ because that is the name in the ‘‘From...’’ line in the message I received. This name is the equivalent of a return address on a mailed envelope.

The mail program lets you prepare mail messages by taking your input from the keyboard. You simply type the text of your message. If you make a mistake on a line, use a combination of backspaces and retyping to correct the mistake before pressing the Enter or Return key. You cannot go back once you have moved to new line. End your input by typing "." (a dot) on a line by itself followed by Return. (Note: some UNIX mail programs don't recognize the dot to end input. If you press dot and Return and find that your program is waiting for more input, type Control-D to terminate input).

If your mail session is interrupted, or if you attempt to send mail to a non-existent log-in name, the text of your message may be preserved in a file called ‘‘dead.letter’’ in your home directory. You can edit the file and then send it using the method described next.

You can also prepare mail messages ‘‘off-line’’ by using your favorite editor. (If your editor is a word processor, be sure to create a plain ASCII text file.) This method gives you more control over the message appearance because you can go back and edit the message as much as you want before sending it. After preparing the message and saving it in a file, you send it by invoking the mail program command with command-line redirection.

Listing 1D shows how I might send a prepared message that is contained in the file named ‘‘new_procs’’. The input redirection symbol ‘‘<’’ tells the UNIX system to obtain input for the mail program from a source other than the keyboard, which is the default input device. In this case, the source is the file named ‘‘new_procs’’.

The mail command shown in Listing 1D also shows a new wrinkle: mailing to a list of recipients rather than just one. The ‘‘new_procs’’ file might contain new operating procedures for a project that involves several users. By giving the mail program a list of log-in names, I need to issue only a single command to send the message to all users in the list instead of having to send mail to each user individually.

E-Mail to Remote Systems

[*Editor’s note:* UUCP is no longer widely used for e-mail delivery to remote systems. Rather, now most mail user agents make a direct network connection to the receiving system over the Internet and exchange control information and data using SMTP (Simple Mail Transport Protocol).]

So far we have dealt with sending mail to users on the same system. If your system can communicate with other UNIX systems, you may also be able to exchange mail messages with users at remote locations.

The key to exchanging mail with remote systems is the availability of UNIX-to-UNIX Copy Programs. UUCP is a set of programs that allow UNIX systems to communicate with each other. If your system is able to call other systems or receive calls from them, you can probably exchange e-mail with users outside your system.

To see whether your system is connected via UUCP to other sites, type "uuname". It will print a list of system names, one per line, that your system can call. If you are lucky, one or more of these systems might be a ‘‘backbone’’ site in the UUCP-based network that spans much of the world. Or, at least you might have access to another system that connects to a backbone site either directly or through a short series of intermediate sites. You may need to enlist the aid of a knowledgeable user or system administrator at your site to find out about your system’s connections.

If your system is connected to others, it makes calls on your behalf. You don’t need to know any telephone numbers, because the system has them in a file that is read by UUCP when it is asked to send mail.

Listing 2A depicts a fictional UNIX network with six sites and shows their connections. The backbone sites have connections (not shown) to many other sites in a larger network. Listing 2B shows some sample mail paths that would be used to address mail to some users in the example.

The exclamation point (usually called a ‘‘bang’’) separates entries in the path specification. The initial series of entries is a sequence of system names of sites that relay the message to the user. The user’s log-in name is the last entry in the path and is preceded by the name of the user’s system. You may have to do some digging to find the information you need to form correct mail paths. If UNIX users send you mail, the message will contain the path by which it came. You can usually reverse the sequence of names to reply to the sender.

The standard UNIX mail program is functional and reliable, but it lacks many convenience features that users of Berkeley UNIX have come to expect. The Berkeley (capital M) mail command has many helpful features to enhance all aspects of receiving, preparing, and sending mail.

A more recent System V UNIX mail program called ‘‘mailx’’ is very similar to Berkeley ‘‘Mail’’ and is available on some AT&T UNIX systems. On some UNIX and XENIX systems, the command named ‘‘mail’’ is a link (another name) to either ‘‘Mail’’ or ‘‘mailx’’. In the following descriptions, I will use the command named ‘‘mailx, but the material applies to the Berkeley MUA with little or no change.

One of the ‘‘mailx’’ features of is the use of a file called ‘‘.mailrc’’ to customize the program to your preferences. When you start this mail user agent, it reads commands and other information from the ‘‘.mailrc’’ file (in your home directory) and configures itself to your specifications.

You have control over various options by using ‘‘set’’ (and ‘‘unset’’) commands. For example, to select the Emacs editor for use in creating mail messages, add the line "set VISUAL=/usr/lbin/emacs" to your ‘‘.mailrc’’ file. The default visual editor is Vi unless your system administrator overrides this choice in a system-wide configuration file. (Note: use a correct path name for the editor of your choice in the `set VISUAL` directive.)

You can also create aliases, which are alternate names that ‘‘mailx’’ translates into full mail addresses. Adding the directive "alias snowman site2!backa!alaska!gus" to your ‘‘.mailrc’’ file lets you use the name ‘‘snowman’’ in place of the full mail path when sending mail. This saves you some time and reduces the risk of typing errors.

Another nice ‘‘mailx’’ feature is the reply function. After you read a message, you may want to send a reply to the sender of the message. Type "R" to reply only to the sender or "r" to reply to the sender and anyone else (except yourself) who received the message. The ‘‘mailx’’ command puts you in the input mode and waits for you to type the reply message. If you want to use a visual editor to create your input, type "~v" (tilde then lowercase ‘‘v’’) and press the Return key. When you have finished creating your reply, quit the editor and type dot (or Control-D) to send the reply.

The ‘‘mailx’’ program has far too many commands and options to cover here. When you decide that you need to know more about this program (and you will), check the documentation provided with you system. You might also want to get the book *UNIX Communications*, by Bart Anderson et al., published by Howard W. Sams, 1987. It offers in-depth coverage of e-mail, UUCP, and other UNIX communications topics.

[*Editor's Notes:* Internet e-mail is simplier to address because you don't have to worry about a path through a set of machines, just the destination machine. For instance, Listing 2C shows the equivalent Internet-style addresses for the UUCP-style addresses shown in Listing 2B.

Also see *Using Email Effectively* by Linda Lamb & Jerry Peek, 1st Edition April 1995 ISBN: 1-56592-103-8, which has a descriptive Web page (http://www.ora.com/www/item/usemail.html).]

Edited by Becca Thomas / Online Editor / UnixWorld Online / editor@unixworld.com

Go To Content   Search

Text entry and HTML markup by Arlene Lee
Last Modified: Sunday, 14-Apr-96 16:48:24 PDT