E-MAIL

MAIL

mail is a comfortable, flexible, interactive program for composing, sending and receiving electronic messages. While reading messages, mail provides you with commands to browse, display, save, delete, and respond to messages. While sending mail, mail allows editing and reviewing of messages being composed, and the inclusion of text from files or other messages.

Incoming mail is stored in the system mailbox for each user (/var/mail/username).

Syntax

Mail may be started by using following syntax at the console prompt

mail [-(options)] /to view mail

mail [-(options)] recipient ... /to send mail

Options

If no recipient is specified, mail attempts to read messages from the system mailbox.

-H Print header summary only.

-N Do not print initial header summary.

-f [filename] Read messages from filename instead of system mailbox. If no filename is specified, the mbox is used.

Starting Mail

You can use the mail command to send a message directly by including names of recipients as arguments on the command line. When no recipients appear on the mail command line, it enters command mode, from which you can read messages sent to you. If you list no recipients and have no messages, mail prints the message: 'No mail for username' and exits. When in command mode (while reading messages), you can send messages using the mail command.

Sending Mail

While you are composing a message to send, mail is in input mode. If no subject is specified as an argument to the command a prompt for the subject is printed. After entering the subject line, mail enters input mode to accept the text of your message to send.

As you type in the message, mail stores it in a temporary file. To review or modify the message, enter the appropriate tilde escapes, listed below, at the beginning of an input line.

To indicate that the message is ready to send, type a dot (or EOF character, normally CTRL-D) on a line by itself. Mail submits the message for routing to each recipient.

Tilde Escapes

The following tilde escape commands can be used when composing messages to send. Each must appear at the beginning of an input line with the escape character (~). The escape character can be entered as text by typing it twice.

~. Simulate EOF (terminate message input).

~? Print a summary of tilde escapes.

~e Invoke the editor to edit the message. The name of the editor is listed in the EDITOR variable. The default editor is ex(1).

~f [message-list] Forward the listed messages, or the current message being read. Valid only when sending a message while reading mail; the messages are inserted without alteration (as opposed to the ~m escape). ~h Prompt for the message header lines: Subject, To, Cc, and Bcc. If the header line contains text, you can edit the text by backspacing over it and retyping.

~p Print the message being entered.

~r filename ~< filename ~<! shell-command

Read in text from the specified file or the standard output of the specified shell-command.

Message are listed and referred to by number. There is, at any time, a current message, which is marked by a > in the header summary. For commands that take an optional list of messages, if you omit a message number as an argument, the command applies to the current message.

A message-list is a list of message specifications, separated by SPACE characters, which may include:. The current message.

n Message number

n. ^ The first undeleted message.

$ The last message.

+ The next undeleted message.

- The previous undeleted message.

* All messages.

n-m An inclusive range of message numbers. user All messages from user. /string All messages with string in the subject line (case ignored).

:c All messages of type c, where c is one of:

d deleted messages

n new messages

o old messages

r read messages

u unread messages

Note: the context of the command determines whether this type of message specification makes sense.

Additional arguments are treated as strings whose usage depends on the command involved. Filenames, where expected, are expanded using the normal shell filename-substitution mechanism.

Commands

While in command mode, if you type in an empty command line (a RETURN or NEWLINE only), the print command is assumed. The following is a complete list of mail commands:

= Print the current message number.

? Print a summary of commands.

| command such as "munpack" or "lp"

quit Exit from mail storing messages that were read in the mbox file and unread messages in the system mailbox. Messages that have been explicitly saved in a file are deleted unless the variable keepsave is set.

reply [message-list], respond [message-list], replysender [message-list]

Send a response to the author of each message in the message-list. The subject line is taken from the first message. If record is set to a filename, a copy of the reply is added to that file. Reply [message]/ Respond [message]/ replyall [message] allows a user to respond to the specified message, including all other recipients of that message. If the variable record is set to a filename, a copy of the reply added to that file. If the replyall variable is set, the actions of Reply/Respond and reply/respond are reversed. The replyall command is not affected by the replyall variable, but always sends the reply to all recipients of the message.

retain Add the list of header fields named to the retained list. Only the header fields in the retain list are shown on your terminal when you print a message. All other header fields are suppressed. The set of retained fields specified by the retain command overrides any list of ignored fields specified by the ignore command. The Type and Print commands can be used to print a message in its entirety. If retain is executed with no arguments, it lists the current set of retained fields.

save [message-list] [filename]

Save the specified messages in the named file. The file is created if it does not exist. If no filename is specified, the file named in the MBOX variable is used, mbox in your home directory by default. Each saved message is deleted from the system mailbox when mail terminates unless the keepsave variable is set. See also the exit and quit commands.

Save [message-list]

Save the specified messages in a file whose name is derived from the author of the first message. The name of the file is taken from the author's name, with all network addressing stripped off. See also the Copy, followup, and Followup commands and the outfolder variables.

shell Invoke the interactive shell listed in the SHELL variable.

top [message-list]

Print the top few lines of the specified messages. If the toplines variable is set, it is taken as the number of lines to print. The default number is 5.

undelete [message-list]

Restore deleted messages. This command only restores messages deleted in the current mail session. If the autoprint variable is set, the last message restored is printed.

write [message-list] [filename]

Write the given messages onto the specified file, but without the header and trailing blank line. Otherwise, this is equivalent to the save command.

Mpack munpack

munpack - unpack messages in MIME or split-uuencode format

munpack [-f] [-q] [-t] [-C directory] [filename ...]

DESCRIPTION

The munpack program reads each RFC-822 message filename and writes all non-text MIME parts or split-uuencoded files as files. If no filename argument is given, munpack reads from standard input.

If the message suggests a file name to use for the imbedded part, that name is cleaned of potential problem characters and used for the output file. If the suggested filename includes subdirectories, they will be created as necessary. If the message does not suggest a file name, the names "part1", "part2", etc are used in sequence.

If the imbedded part was preceded with textual information, that information is also written to a file. The file is named the same as the imbedded part, with any filename extension replaced with ".desc".

mpack - pack a file in MIME format

mpack [-s subject] [-d descriptionfile] [-m maxsize] [-c content-type] file address ...

mpack [-s subject] [-d descriptionfile] [-m maxsize] [-c content-type] -o outputfile file

DESCRIPTION

The mpack program encodes the the named file in one or more MIME messages. The resulting messages are mailed to one or more recipients, written to a named file or set of files.

pine

pine is an application that sends and receives electronic mail to and from anywhere on the Internet. While it does not have a GUI, it has a menu-driven interface, and thus, may be easier to use for novice users. It is started by typing "pine" at a window prompt.

When invoked, the menu of available choices is displayed on the screen with the arrow that points to the first entry. The arrow may be moved up or down by arrow keys of the keyboard keypad. Chose the desired entry by pressing enter key. There are also keys and combinations of keys on the bottom of the screen with the short explanation of their functions. Notice "other cmds" as one of the choices. Pressing it will show you some additional keys and combinations.

The figure below shows the initial pine menu.

See Pico in the Editing section for additional information on Pine's editor.

Pine also supports MIME, The Multipurpose Internet Mail Extensions defined in RFC-1521. This allows Pine to send and receive multipart and multimedia e-mail. Pine meets the minimal MIME compliance requirements and is able to view most parts of any received MIME message and to save all parts to files, whatever their format. On the composing side, the focus of the MIME implementation has been to allow users to attach files to messages so they can transfer arbitrary messages, rather than on creating true multi-media e-mail with graphics and sounds. This allows UNIX Pine and PC-Pine users to mail spread sheets and other such files back and forth. Pine will recognize a few of the multimedia formats such as GIF files. When they are attached they are tagged as being images and if Pine is running under openwin it will call xv to display them.

TALK

talk is a visual communication program which copies lines from your terminal to that of another user.

Syntax

talk username [ttyname]

If you wish to talk to someone on your own machine, then username is just the person's login name. If you wish to talk to a user on another host, then username is one of the following forms:

host!user
host.user
host:user
user@host (the last one is perhaps preferred)

If you want to talk to a user who is logged in more than once, the ttyname argument may be used to indicate the appropriate terminal name.

talk yourname@yourmachine

It does not matter from which machine the recipient replies, as long as their login name is the same. Once communication is established, the two parties may type simultaneously, with their output appearing in separate windows. Typing CTRL-L redraws the screen, while your erase, kill, and word kill characters will work in talk as normal. To exit, just type your interrupt character; talk then moves the cursor to the bottom of the screen and restores the terminal.

Permission to talk may be denied or granted by use of the mesg command. At the outset talking is allowed.

Write

write user [terminal]

The write utility reads lines from the user's standard input and writes them to the terminal of another user. When first invoked, it writes the message:

Message from sender-login-id (sending-terminal) [date]...

to user. When it has successfully completed the connection,the sender's terminal will be alerted twice to indicate that what the sender is typing is being written to the recipient's terminal.

If the recipient wants to reply, this can be accomplished by typing

write sender-login-id [sending-terminal]