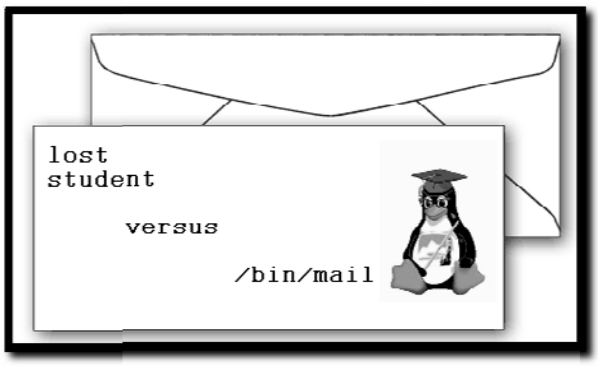


Understanding /bin/mail : Lost Student versus /bin/mail

By: Michael Wicherski

My thanks to:

Mr. Simms

(Also a big thanks to whoever designed that version of the Linux penguin I used)

Understanding /bin/mail - Contents

Title and Introduction

The title you already know, Understanding /bin/mail : Lost Student versus /bin/mail

This guide is written to allow you to master the use of the Linux mail system and apply it for anything you wish, be it class work or personal means of communication.

NOTE This guide is written with the understanding that you have everything required to complete the task (see "What You Will Need" section), and that you are operating on the Opus server of Cabrillo College; However I would like to point out you may apply this to any Linux system supporting this program.

What You Will Need

The following items are what you will require to be able to use the /bin/mail program on Opus servers at Cabrillo College:

- An Opus server account, which will be provided to you by your professor or may be obtained at the Computer Technology Center (CTC)
- Either local access to the Opus server at the CTC or a remote way of accessing the server. The most popular of accessing the server remotely is by using PuTTY. For information on how to configure PuTTY, see here (also contains a link to the download site).
- A basic understanding of Linux and how to operate a terminal interface.
- *NOTE* A glossary and a man info page has been provided at the end of this tutorial for your convenience.

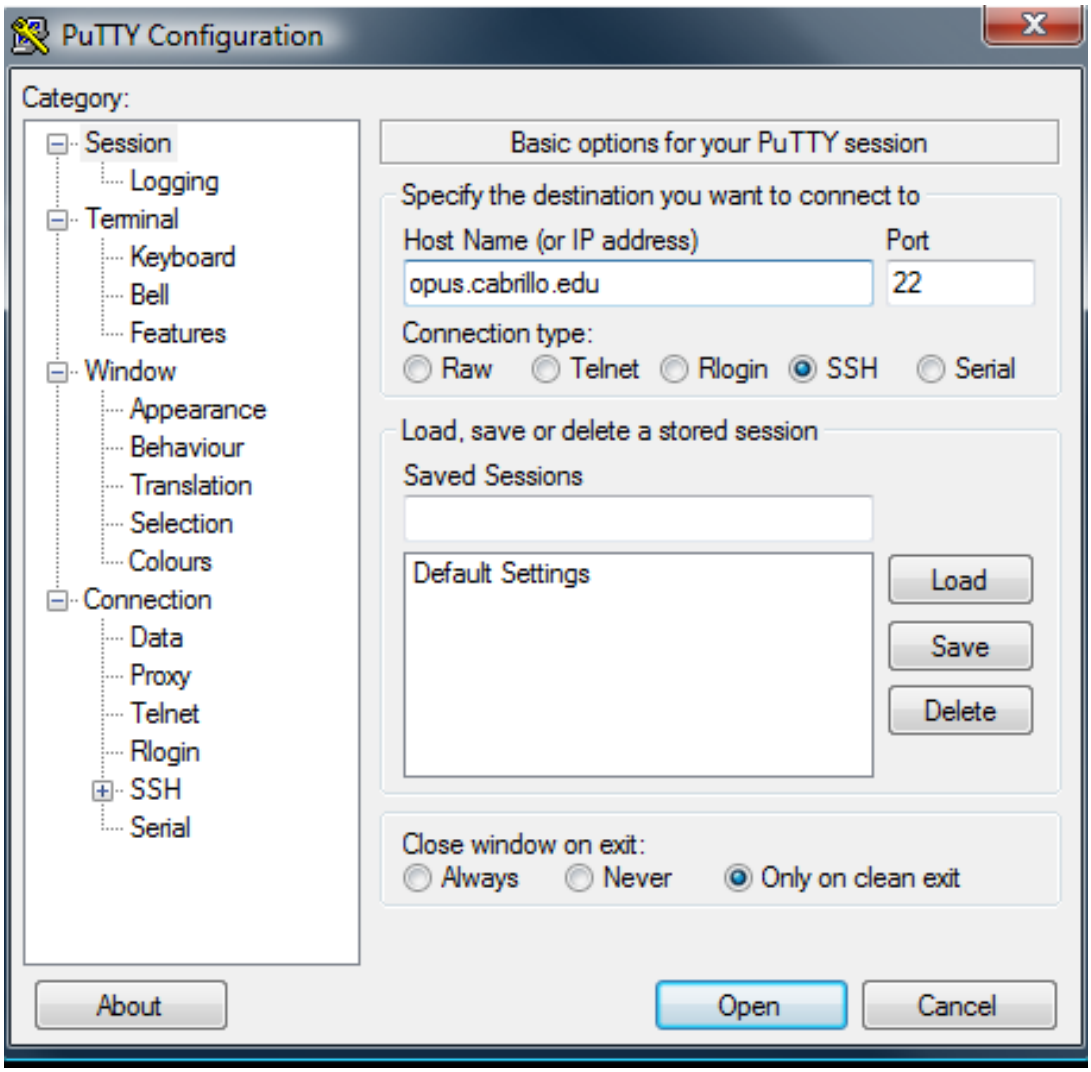
Logging Into Opus

There are two ways to login to the Opus server as mentioned before: Remotely and Locally.

If you are going to be connecting remotely using PuTTY read the following section, otherwise proceed to "Connecting Locally"

Remotely - Using PuTTY

1. Open up PuTTY and connect to the Opus server.



2. Login using your Opus account username and then password, which you will be prompted for

   Login as: _____ (just your 8 character username)
   username@opus.cabrillo.edu's password: _____

3. At the next prompt just hit enter or return (if you have an ancient keyboard ~.^)

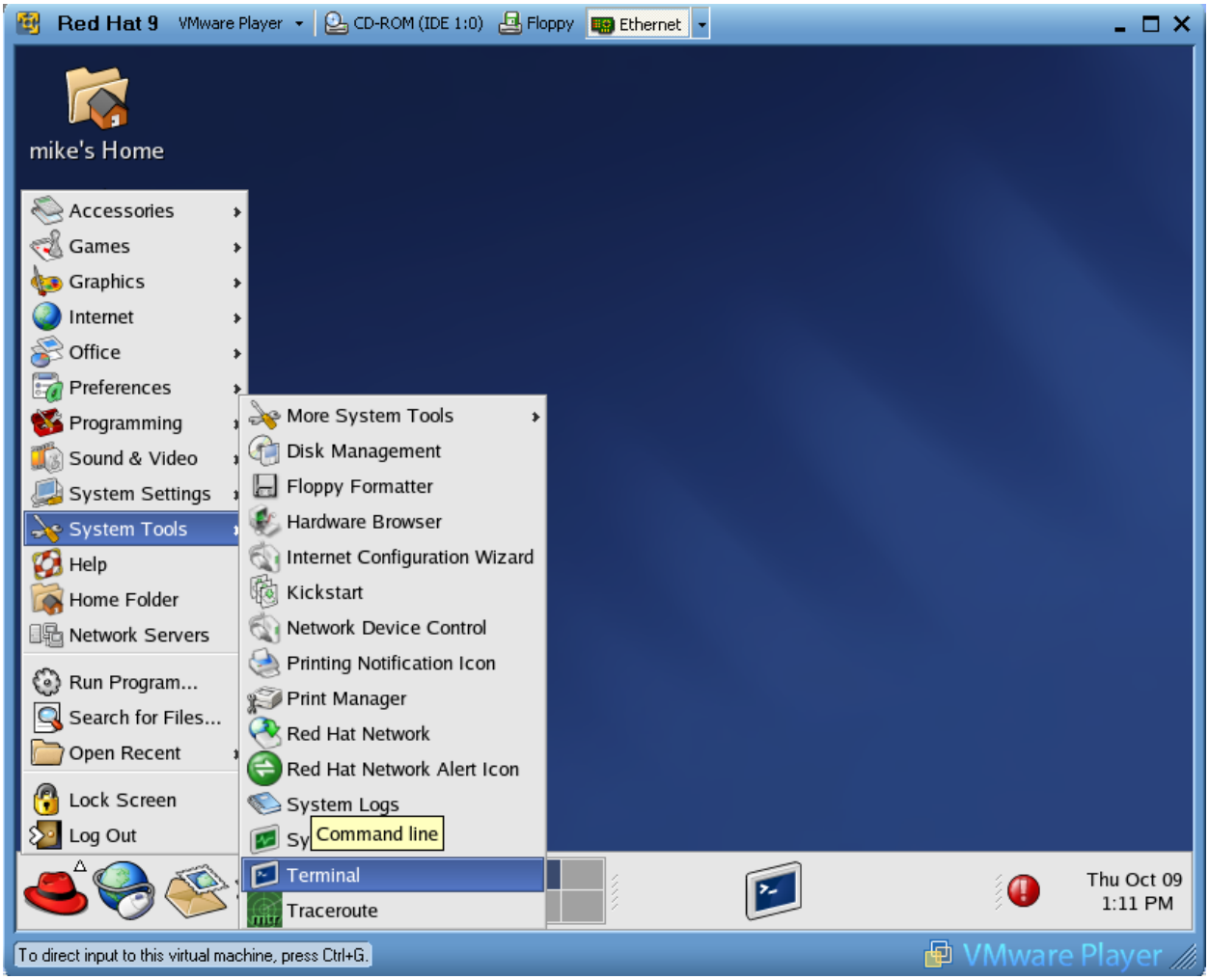
   Terminal type? [xterm] enter
   Terminal type is xterm. (just tells you default terminal and you now see your prompt)

   /home/<class>/<username> $

Locally – Using VMware or actual Linux install

(For the purpose of this tutorial we will be using the Red Hat distribution of Linux as it is available through the Cabrillo College system in the labs, as well as VMWare solutions.)

1. Open a console aka terminal: click the redhat and continue the directory path like highlighted below



2. You will get a terminal window with the following prompt:

   [cisco@localhost cisco] $

3. Type in the ssh command as well as your username to connect to the opus server:

   [cisco@localhost cisco] $ ssh username@opus.cabrillo.edu

4. When prompted for your password by the following prompt, enter it

   username@opus.cabrillo.edu's password: _____ (you are now logged in)

Starting up and using /bin/mail

Ok. We are finally ready to use mail.

To start up the program simply type “mail” at your prompt

    /home/<class>/<username> $ mail

More than likely you will see a message that says

    “No mail for <username>”

And be returned to your prompt

Otherwise, if you have mail and are now in the mail program read further to find out how to exit (or read)

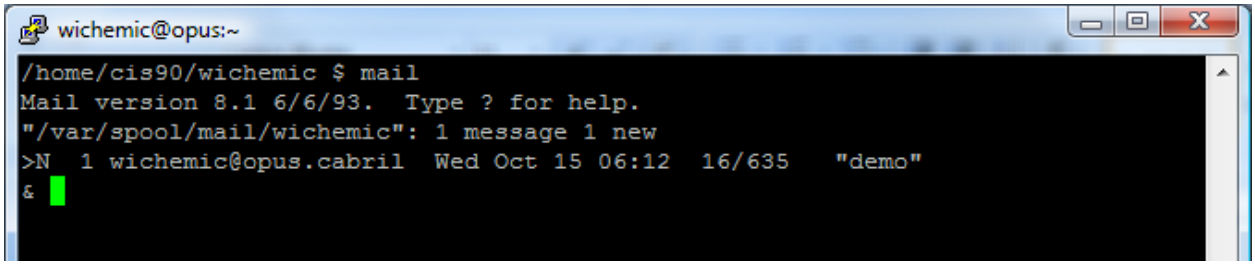
NOTE Make sure you are in your home directory. That means there is nothing after <username> in your prompt except the $ like shown above, otherwise it won’t work. (well some functions won’t; just better to be in home directory to avoid problems later)**

Receiving

You will be informed when you login if you have received any new mail, so no problem there.

If you want to manually check if it has arrived, then simply start up mail like we did earlier.

Reading



We now see that when mail was started, I was told its version number and that you can type "?" for help, which will display all the commands available to you.

It also tells me the location of my "unread" mail which is that ugly thing /var/spo…

Now finally on to my messages.

You can see that I have 1 new message that I have not read (from myself for the purpose of this tutorial)

The text you see:

>N 1 wichemic@opus.cabril Wed Oct 15 06:12 16/635 "demo"

>N	means the message is new and unread
1	is the messages number (if I had more, it would progress to 2,3,…,n)
w….	is the sender's address (again, I sent it to myself, so it is my address; notice the address gets cutoff because it is too long)
…	the date and time
…	don't worry about these numbers, aren't important for the purpose of this tutorial
"x"	anything between the quotes is the subject, here it is "demo"

To read your mail, simply type its number in at the prompt and mail will display the contents:

& 1

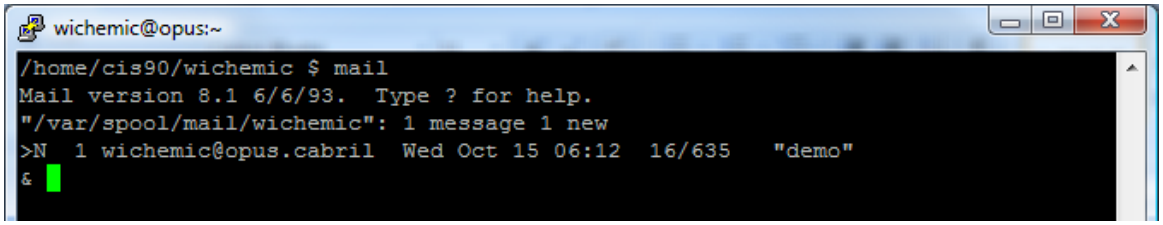
That would open message 1 (the only one in my inbox at the moment)

Exiting

An important thing to note with mail is that you can exit the program in one of two ways.

Typing both “q” and “x” will exit you from the mail program, but with very different results.

For now, let’s stick to exiting with “x”, see “Saving” for more info about “q”.



As you can see, it is a different prompt than usual, it is simply “ & “ which lets you know you are no longer interfacing with the shell, but instead inside a program.

So to exit with “x” :

& x

That’s all there is to it.

Saving

There are two ways you can save your mail. One is by letting the system take care of it for you into a default directory called mbox; the other is you specify where you want your mail saved.

- If you simply exit by typing “q”, then all the mail you have read gets moved to mbox.

- If you want to save a message to a specific “folder” (it’s really a file, but I won’t go into detail on that here), then you must type

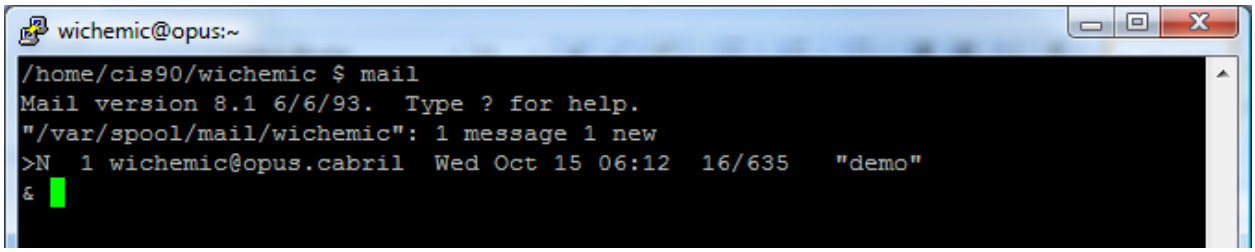
  s # <filename>

  where this means:

  s save

  # the number of the message you want to save

  <filename> where you want this message saved



Recall the different parts of a message listing (covered here)

So to save message 1, type

& s 1 ReadMail

This will save message 1 to the “folder” titled “ReadMail”

Note: You can save multiple messages by using a space to separate message numbers:

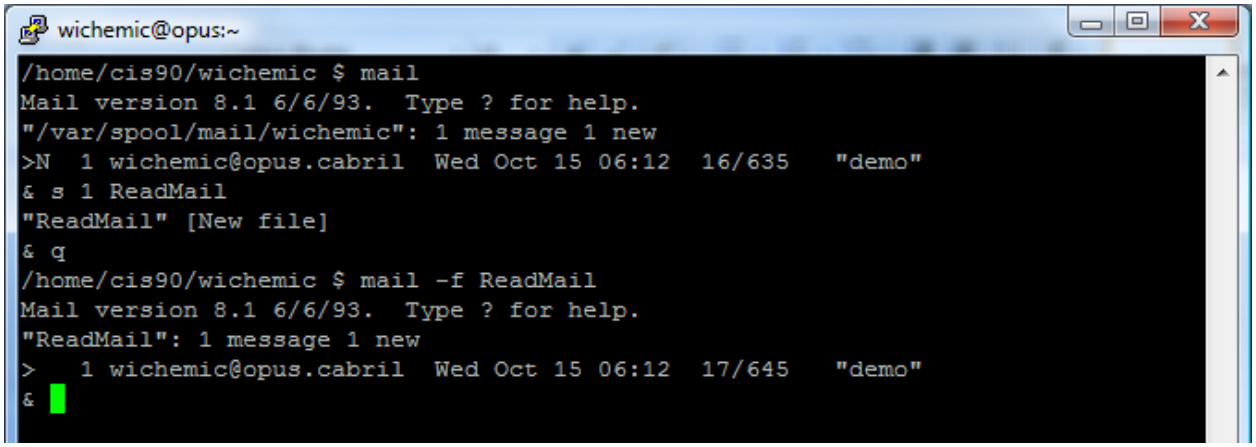
& s 1 2 3 4 5 ReadMail

Would save messages 1,2,3,4,5 to ReadMail

Reading Saved Mail

Well great, now we know how to save messages. But just typing “mail” at the prompt and starting the mail program won’t let you read them.

The solution is to type mail –f <folder> (or just mail –f if you want it to go to the default folder “mbox”
note, mbox is where all your messages get stored if you exit mail using “q” after you have read them

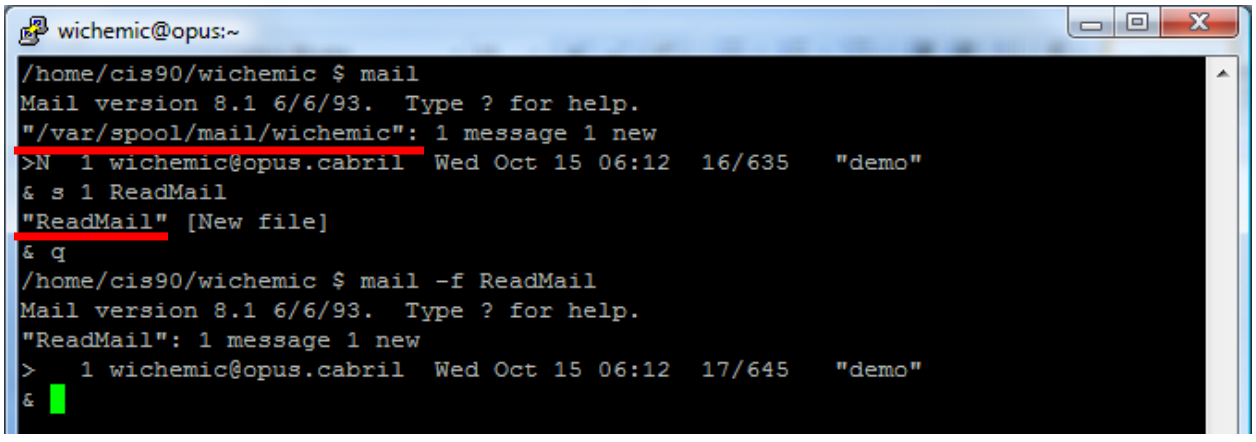


In this screenshot you can see that I have saved message # 1 (s 1 ReadMail) in my inbox to ReadMail, after which I exited mail using q (you cannot navigate different boxes from within mail).

After which I opened the ReadMail folder by typing

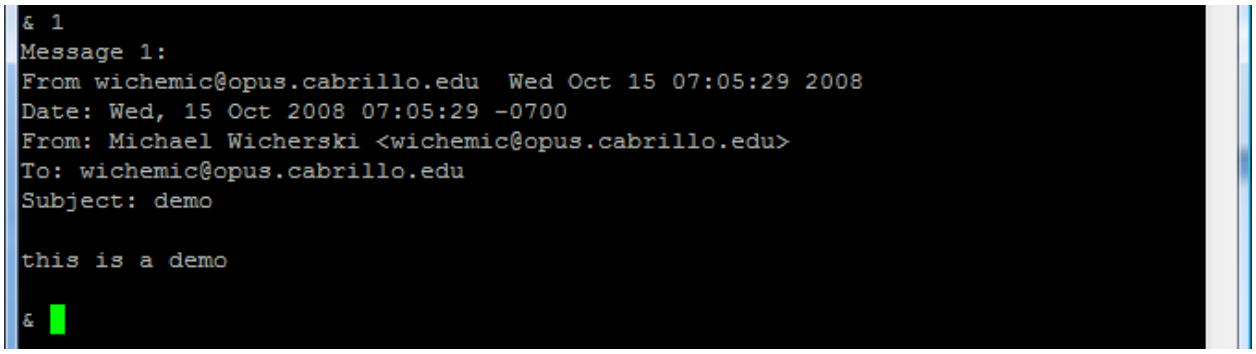
& mail –f ReadMail

Notice how mail shows you the folder you are in; inbox (“/var/… “ versus “ReadMail”)



Reading mail is the same for saved messages once you are inside their containing folder, just type the # you want to read.

Layout of a Mail Message



So when you type 1 at the prompt to read the demo message, (which I had to resend myself, so the time is different than before; don't mind it), you get the above readout

From who, when, subject, contents, and you get thrown back to the prompt

Finding a User to send mail to

Several ways to do this:

- Lean over and beg the person next to you for their username
- Memorize everyone's first and last name (first 5 letters of last name first 3 of first name)
- Type "who" at the main prompt (not mail prompt) and get a BARRAGE of usernames of everyone connected to the server
- OR could be very cool and sly about it, and use the following trick (I am of course assuming that you are using this for school purposes at therefore are in a class or group)
- **Note: you **CAN** send mail to "the outside world" aka something like bart@yahoo.com, but you **CANNOT** send mail to an opus account from a yahoo account for example.**

The trick to find someone to mail in your class/group.

If I lose you, I'm sorry, read over it a few times; It should make sense.

Ok so, exit mail; since you cannot send from within mail(don't ask).

Now look at your prompt.

/home/<class>/<username> $

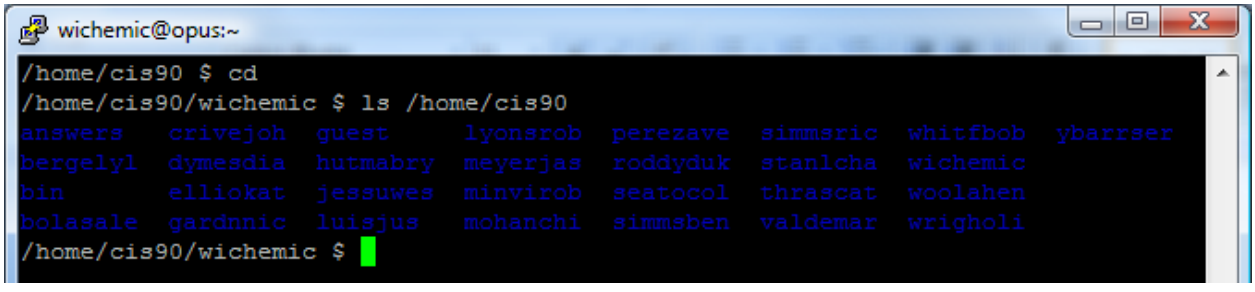
This means that your home folder (where you are currently) is in the folder called "<class>" which is inside a folder called "home". Now notice, that your username is actually your own folder, which means that everyone else in your class, also has a folder in the <class> folder with their username on it!

To get to that information, type

/home/<class>/<username> $ ls /home/<class>

What that does is list EVERY folder (except hidden ones, don't worry) in the <class> folder

Guess what, you now have the username of everyone in your class!



(Notice they are blue, because they are folders)

Composing and Sending

Ok. So now we are ready to send our very first ever message!

Ok, so you are at your prompt

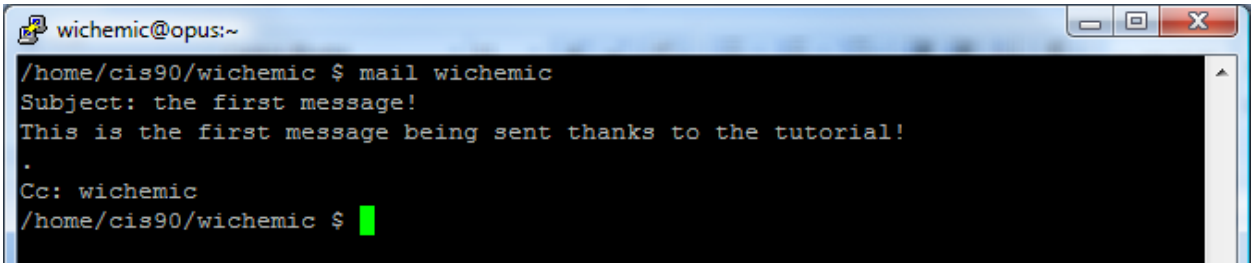
/home/<class>/<username> $

1. To mail someone you type mail <username>

   /home/<class>/<username> $ mail <username>

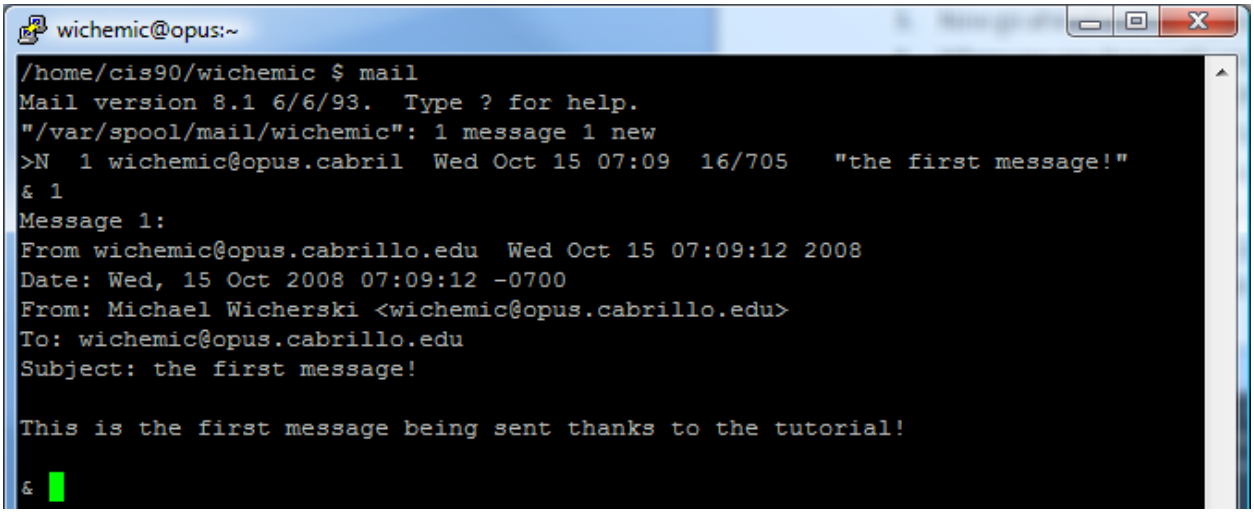
Note: You can mail multiple people with mail <username> <username> <username>

2. The next thing that will come up is a line prompting for the subject, so type one in.
3. Now go ahead and type your message
4. When you are done with your message you can "finish" it two ways:
   a. Pressing ctrl+D (EOF: End of File command)
   b. Or by hitting enter to get a new line and typing ONLY a period on it and hitting enter again
5. The next thing that comes up is a " CC: " option (**Note: this is the only way to send yourself a copy, so it is good practice to input your username here**)
6. Once you hit enter on the CC: (which you can leave blank), the message is sent; and no, unfortunately you don't get any confirmation, which is why CC: yourself is a good idea



<sigh> Spamming up my own inbox again… The things I do for you guys ☺

Note: Although I CC myself, since it is addressed to me, I only get 1 copy.



Replying

Replying. Commonly used!

Two ways to do it:

- to sender
- to sender and all recipients.

Here is where it gets slightly confusing. Linux is cASe sEnSiTIve.

Recall how we saved messages using "s"

To reply to the **sender only**

& R #

To reply to **sender and all recipients**

& r #

Aka

& r 1

Would reply to the sender of the message #1 (in the current box I'm in (see saving)) as well as everyone else he sent the message to.

Whereas

& R 1

Would reply only to the sender of message #1

Once you execute a reply command, mail automatically does the subject and addressing for you

(subject is just RE:<subject>)

So you can just type your reply message and treat it as if you were sending a normal message from there on out.

Forwarding Messages

Forwarding message is such a breeze if you know how to reply or save a message already!

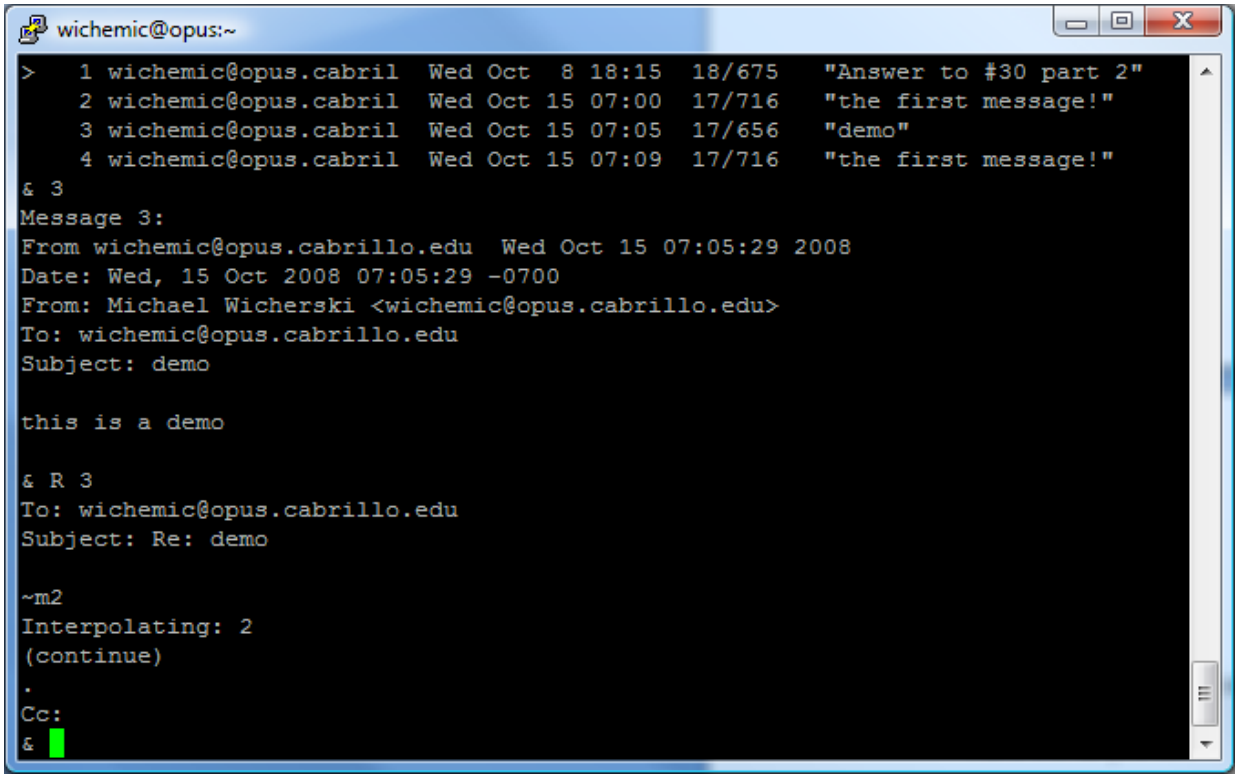
Here's what you do!

1. Either start a new message or reply to an existing one
2. In the body of the message, hit enter for a new line
3. Type "~m#" and hit enter (no spaces)
4. You will get a message saying "Interpolating: #"
5. Then on a new line "(continue)"
6. That is your cue that the message you wanted to forward has been copied into the message you are composing and you can continue typing, or finish up and mail it!

~ command character

m "message"

number of message in the box you are currently in



You can see I forwarded message 2 by replying to message 3, using a capital R which means I sent it only to the sender (which in this case doesn't matter because I was both sender and recipient haha)

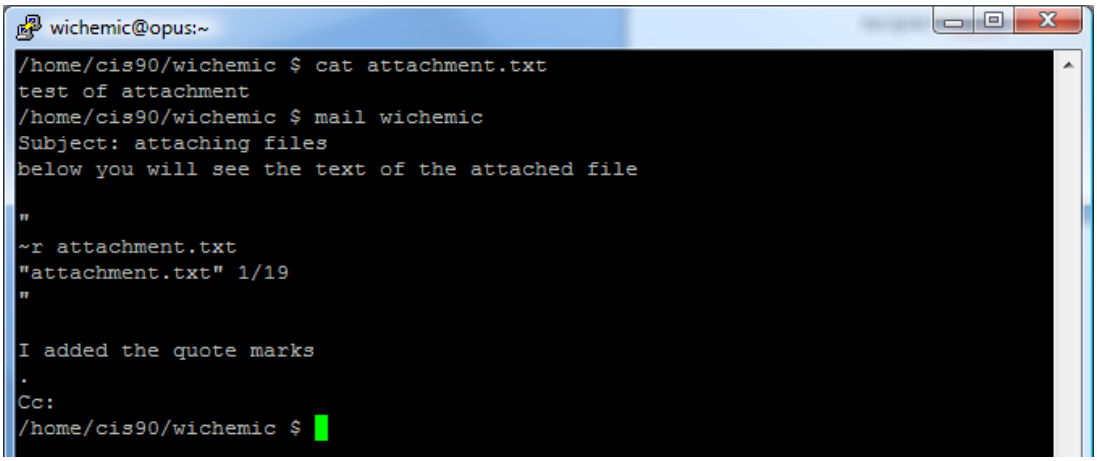
Attachments

As far as we can tell, only text file attachments are possible using /bin/mail. By we, I mean myself, the professor, and the rest of the class.

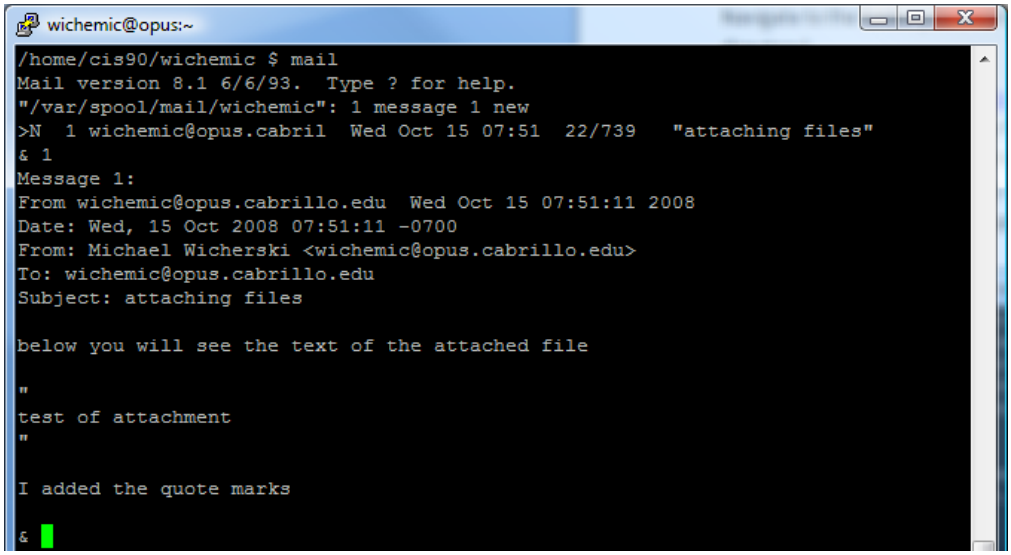
To attach a file, you pretend you are forwarding the file.

Navigate to the folder you have the file stored in (for sanity's sake, I'll presume that will be your home directory)

1. Start a message or reply to one, whichever;
2. Now just as if you were forwarding a message using ~m#
3. Use ~r <filename> (notice there is a space between the ~r and <filename>; a little different)
4. The computer will reply with the name of the file in quotes and your cursor will be on the line below that, continue typing your message, or send



Above you can see the contents of the attachment.txt file "test of attachment" and how I attached it



< Here you can see what the delivered message looks like

Glossary of Terms

Note, these are lament descriptions, and not “technically” precise, they exist to give the “new” user an general idea of what is going on

- CC: - carbon copy
- console - see terminal
- interpolating - joining
- ls command - “list” command, type ls and the directory to list its contents, type just ls to list current directory
- mail - /bin/mail a program for sending and receiving mail on Linux systems
- man - a command that will give you information about any command type “man <command>” example: “man mail”
- Opus - Cabrillo College’s Linux Server
- Prompt - what the shell (computer) prints to you every time it is waiting for a command
- PuTTY - a Secure Shell client which allows for connecting to servers
- terminal - interface which one uses to interact with a text command based system

Man Reference Page for /bin/mail

**Note, obviously I did not type all this out; so the formatting might be a bit off due to the copy and paste; if you want to view the original, here is the link: http://unixhelp.ed.ac.uk/CGI/man-cgi?mail

---

MAIL(1) BSD General Commands Manual MAIL(1)

NAME

mail, mailx, Mail - send and receive mail

SYNOPSIS

```
mail [-eIinv] [-a header] [-b bcc-addr] [-c cc-addr] [-s subject] to-
addr
     [...] [-- sendmail-options [...]]
mail [-eIiNnv] -f [name]
mail [-eIiNnv] [-u user]
```

DESCRIPTION

mail is an intelligent mail processing system which has a command syntax reminiscent of ed(1) with lines replaced by messages.

The options are as follows:

-a Specify additional header fields on the command line such as "X-Loop: foo@bar" etc. You have to use quotes if the string contains spaces. This argument may be specified more than once, the headers will then be concatenated.

-b bcc-addr
Send blind carbon copies to bcc-addr.

-c cc-addr
Send carbon copies to list of users. cc-addr should be a comma separated list of names.

-e Don't send empty mails. If the body is empty skip the mail.

-f [name]
Read in the contents of your mailbox (or the specified file name) for processing; when you quit, mail writes undeleted messages back to this file.

-I Forces mail to run in interactive mode, even when input is not a terminal. In particular, the special ~ command character, used when sending mail, is only available interactively.

-i Ignore tty interrupt signals. This is particularly useful when

using mail on noisy phone lines.

-N Inhibits initial display of message headers when reading mail or editing a mail folder.

-n Inhibits reading /etc/mail.rc upon startup.

-s subject
Specify subject on command line (only the first argument after the -s flag is used as a subject; be careful to quote subjects containing spaces).

-u user
Equivalent to:

```
$ mail -f /var/mail/user
```

except that locking is done.

-v Verbose mode. The details of delivery are displayed on the user's terminal.

Startup actions

At startup time, mail will execute commands in the system command file, /etc/mail.rc, unless explicitly told not to by using the -n option. Next, the commands in the user's personal command file ~/.mailrc are executed. mail then examines its command line options to determine whether the user requested a new message to be sent or existing messages in a mailbox to be examined.

Sending mail

To send a message to one or more people, mail can be invoked with arguments which are the names of people to whom the mail will be sent. You are then expected to type in your message, followed by a control-D ('^D') at the beginning of a line. The section below, Replying to or originating mail, describes some features of mail available to help you compose your letter.

Reading mail

In normal usage, mail is given no arguments and checks your mail out of the post office, then prints out a one line header of each message found.
The current message is initially set to the first message (numbered 1) and can be printed using the print command (which can be abbreviated p). Moving among the messages is much like moving between lines in ed(1); you may use + and - to shift forwards and backwards, or simply enter a message number to move directly.

Disposing of mail

After examining a message you can delete (d) or reply (r) to it. Deletion causes the mail program to forget about the message. This is not irreversible; the message can be undeleted (u) by giving its number, or the mail session can be aborted by giving the exit (x) command. Deleted

messages, however, will usually disappear, never to be seen again.

Specifying messages

Commands such as print and delete can be given a list of message numbers as arguments to apply to a number of messages at once. Thus delete 1 2 deletes messages 1 and 2, while delete 1-5 deletes messages 1 through 5. The special name '*' addresses all messages and '$' addresses the last message; thus the command top which prints the first few lines of a message could be used in top * to print the first few lines of all messages.

Replying to or originating mail

You can use the reply command to set up a response to a message, sending it back to the person who it was from. Text you then type in, up to an end-of-file, defines the contents of the message. While you are composing a message, mail treats lines beginning with the tilde ('~') character specially. For instance, typing ~m (alone on a line) will place a copy of the current message into the response, right shifting it by a single tab-stop (see the indentprefix variable, below). Other escapes will set up subject fields, add and delete recipients to the message, and allow you to escape to an editor to revise the message or to a shell to run some commands. (These options are given in the summary below.)

Ending a mail processing session

You can end a mail session with the quit (q) command. Messages which have been examined go to your mbox file unless they have been deleted, in which case they are discarded. Unexamined messages go back to the post office (see the -f option above).

Personal and system wide distribution lists

It is also possible to create personal distribution lists so that, for instance, you can send mail to ``cohorts'' and have it go to a group of people. Such lists can be defined by placing a line like

```
alias cohorts bill ozalp jkf mark kridle@ucbcory
```

in the file .mailrc in your home directory. The current list of such aliases can be displayed with the alias command in mail. System wide distribution lists can be created by editing /etc/aliases, (see aliases(5) and sendmail(8)); these are kept in a different syntax. In mail you send, personal aliases will be expanded in mail sent to others so that they will be able to reply to the recipients. System wide aliases are not expanded when the mail is sent, but any reply returned to the machine will have the system wide alias expanded as all mail goes through sendmail.

Network mail (ARPA, UUCP, Berknet)

See mailaddr(7) for a description of network addresses.

mail has a number of options which can be set in the .mailrc file to alter its behavior; thus set askcc enables the askcc feature. (These options are summarized below.)

SUMMARY

(Adapted from the ``Mail Reference Manual''.)

Each command is typed on a line by itself, and may take arguments follow-
ing the command word. The command need not be typed in its entirety --
the first command which matches the typed prefix is used. For commands
which take message lists as arguments, if no message list is given, then
the next message forward which satisfies the command's requirements is
used. If there are no messages forward of the current message, the
search proceeds backwards, and if there are no good messages at all, mail
types ``No applicable messages'' and aborts the command.

- Print out the preceding message. If given a numeric argument n, goes to the nth previous message and prints it.

? Prints a brief summary of commands.

! Executes the shell (see sh(1) and csh(1)) command which follows.

Print (P) Like print but also prints out ignored header fields. See also print, ignore, and retain.

Reply (R) Reply to originator. Does not reply to other recipients of the original message.

Type (T) Identical to the Print command.

alias (a) With no arguments, prints out all currently defined aliases. With one argument, prints out that alias. With more than one argument, creates a new alias or changes an old one.

alternates
(alt) The alternates command is useful if you have accounts on several machines. It can be used to inform mail that the listed addresses are really you. When you reply to messages, mail will not send a copy of the message to any of the addresses listed on the alternates list. If the alternates command is given with no argument, the current set of alternate names is displayed.

chdir (c) Changes the user's working directory to that specified, if given. If no directory is given, then changes to the user's login directory.

copy (co) The copy command does the same thing that save does, except that it does not mark the messages it is used on for deletion when you quit.

delete (d) Takes a list of messages as argument and marks them all as deleted. Deleted messages will not be saved in mbox, nor will they be available for most other commands.

dp (also dt) Deletes the current message and prints the next message. If there is no next message, mail says ``No more messages.''

edit (e) Takes a list of messages and points the text editor at each one in turn. On return from the editor, the message is read back in.

exit (ex or x) Effects an immediate return to the shell without modifying the user's system mailbox, his mbox file, or his edit file in -f.

file (fi) The same as folder.

folders
List the names of the folders in your folder directory.

folder (fo) The folder command switches to a new mail file or folder. With no arguments, it tells you which file you are currently reading. If you give it an argument, it will write out changes (such as deletions) you have made in the current file and read in the new file. Some special conventions are recognized for the name. # means the previous file, % means your system mailbox, %user means user's system mailbox, & means your mbox file, and +folder means a file in your folder directory.

from (f) Takes a list of messages and prints their message headers.

headers
(h) Lists the current windowful of headers. To view the next or previous group of headers, see the z command.

help A synonym for ?.

hold (ho, also preserve) Takes a message list and marks each message therein to be saved in the user's system mailbox instead of in mbox. Does not override the delete command.

ignore Add the list of header fields named to the ignored list. Header fields in the ignore list are not printed on your terminal when you print a message. This command is very handy for suppression of certain machine-generated header fields. The Type and Print commands can be used to print a message in its entirety, including ignored fields. If ignore is executed with no arguments, it lists the current set of ignored fields.

inc Incorporate any new messages that have arrived while mail is being read. The new messages are added to the end of the message list, and the current message is reset to be the first new mail message. This does not renumber the existing message list, nor does it cause any changes made so far to be saved.

list (l) List the valid mail commands.

mail (m) Takes as argument login names and distribution group names and sends mail to those people.

mbox Indicate that a list of messages be sent to mbox in your home directory when you quit. This is the default action for messages if you do not have the hold option set.

more (mo) Takes a message list and invokes the pager on that list.

next (n) (like + or CR) Goes to the next message in sequence and types it. With an argument list, types the next matching message.

preserve
(pre) A synonym for hold.

print (p) Takes a message list and types out each message on the user's terminal.

quit (q) Terminates the session, saving all undeleted, unsaved messages in the user's mbox file in his login directory, preserving all messages marked with hold or preserve or never referenced in his system mailbox, and removing all other messages from his system mailbox. If new mail has arrived during the session, the message ``You have new mail'' is given. If given while editing a mailbox file with the -f flag, then the edit file is rewritten. A return to the shell is effected, unless the rewrite of edit file fails, in which case the user can escape with the exit command.

reply (r) Takes a message list and sends mail to the sender and all recipients of the specified message. The default message must not be deleted.

respond
A synonym for reply.

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 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 (s) Takes a message list and a filename and appends each message in turn to the end of the file. The filename in quotes, followed by the line count and character count is echoed on the user's terminal.

saveignore
saveignore is to save what ignore is to print and type. Header fields thus marked are filtered out when saving a message by save or when automatically saving to mbox.

saveretain
saveretain is to save what retain is to print and type. Header fields thus marked are the only ones saved with a message when

saving by save or when automatically saving to mbox. saveretain overrides saveignore.

set (se) With no arguments, prints all variable values. Otherwise, sets option. Arguments are of the form option=value (no space before or after =) or option. Quotation marks may be placed around any part of the assignment statement to quote blanks or tabs, i.e., set indentprefix="->".

shell (sh) Invokes an interactive version of the shell.

size Takes a message list and prints out the size in characters of each message.

source The source command reads commands from a file.

top Takes a message list and prints the top few lines of each. The number of lines printed is controlled by the variable toplines and defaults to five.

type (t) A synonym for print.

unalias
Takes a list of names defined by alias commands and discards the remembered groups of users. The group names no longer have any significance.

undelete
(u) Takes a message list and marks each message as not being deleted.

unread (U) Takes a message list and marks each message as not having been read.

unset Takes a list of option names and discards their remembered values; the inverse of set.

visual (v) Takes a message list and invokes the display editor on each message.

write (w) Similar to save, except that only the message body (without the header) is saved. Extremely useful for such tasks as sending and receiving source program text over the message system.

xit (x) A synonym for exit.

z mail presents message headers in windowfuls as described under the headers command. You can move mail's attention forward to the next window with the z command. Also, you can move to the previous window by using z-.

Tilde/escapes

Here is a summary of the tilde escapes, which are used when composing messages to perform special functions. Tilde escapes are only recognized at the beginning of lines. The name ``tilde escape'' is somewhat of a misnomer since the actual escape character can be set by the option

escape.

~!command
Execute the indicated shell command, then return to the message.

~bname ...
Add the given names to the list of carbon copy recipients but do not make the names visible in the Cc: line ("blind" carbon copy).

~cname ...
Add the given names to the list of carbon copy recipients.

~d Read the file dead.letter from your home directory into the message.

~e Invoke the text editor on the message collected so far. After the editing session is finished, you may continue appending text to the message.

~fmessages
Read the named messages into the message being sent. If no messages are specified, read in the current message. Message headers currently being ignored (by the ignore or retain command) are not included.

~Fmessages
Identical to ~f, except all message headers are included.

~h Edit the message header fields by typing each one in turn and allowing the user to append text to the end or modify the field by using the current terminal erase and kill characters.

~mmessages
Read the named messages into the message being sent, indented by a tab or by the value of indentprefix. If no messages are specified, read the current message. Message headers currently being ignored (by the ignore or retain command) are not included.

~Mmessages
Identical to ~m, except all message headers are included.

~p Print out the message collected so far, prefaced by the message header fields.

~q Abort the message being sent, copying the message to dead.letter in your home directory if save is set.

~Rstring
Use string as the Reply-To field.

~rfilename
Read the named file into the message.

~sstring
Cause the named string to become the current subject field.

~tname ...
Add the given names to the direct recipient list.

~v Invoke an alternate editor (defined by the VISUAL option) on the message collected so far. Usually, the alternate editor will be a screen editor. After you quit the editor, you may resume appending text to the end of your message.

~wfilename
Write the message onto the named file.

~|command
Pipe the message through the command as a filter. If the command gives no output or terminates abnormally, retain the original text of the message. The command fmt(1) is often used as command to rejustify the message.

~:mail-command
Execute the given mail command. Not all commands, however, are allowed.

~~string
Insert the string of text in the message prefaced by a single ~. If you have changed the escape character, then you should double that character in order to send it.

Mail options

Options are controlled via set and unset commands. Options may be either binary, in which case it is only significant to see whether they are set or not; or string, in which case the actual value is of interest. The binary options include the following:

append Causes messages saved in mbox to be appended to the end rather than prepended. This should always be set (perhaps in /etc/mail.rc).

ask, asksub
Causes mail to prompt you for the subject of each message you send. If you respond with simply a newline, no subject field will be sent.

askbcc Causes you to be prompted for additional blind carbon copy recip-
ients at the end of each message.Responding with a newline indicates your satisfaction with the current list.

askcc Causes you to be prompted for additional carbon copy recipients at the end of each message. Responding with a newline indicates your satisfaction with the current list.

autoinc
Causes new mail to be automatically incorporated when it arrives. Setting this is similar to issuing the inc command at each prompt, except that the current message is not reset when new

mail arrives.

autoprint
Causes the delete command to behave like dp; thus, after deleting a message, the next one will be typed automatically.

debug Setting the binary option debug is the same as specifying -d on the command line and causes mail to output all sorts of information useful for debugging mail.

dot The binary option dot causes mail to interpret a period alone on a line as the terminator of a message you are sending.

hold This option is used to hold messages in the system mailbox by default.

ignore Causes interrupt signals from your terminal to be ignored and echoed as @'s.

ignoreeof
An option related to dot is ignoreeof which makes mail refuse to accept a control-D as the end of a message. ignoreeof also applies to mail command mode.

keep Setting this option causes mail to truncate your system mailbox instead of deleting it when it's empty.

keepsave
Messages saved with the save command are not normally saved in mbox at quit time. Use this option to retain those messages.

metoo Usually, when a group is expanded that contains the sender, the sender is removed from the expansion. Setting this option causes the sender to be included in the group.

noheader
Setting the option noheader is the same as giving the -N flag on the command line.

nosave Normally, when you abort a message with two interrupt characters (usually control-C), mail copies the partial letter to the file dead.letter in your home directory. Setting the binary option nosave prevents this.

Replyall
Reverses the sense of reply and Reply commands.

quiet Suppresses the printing of the version when first invoked.

searchheaders
If this option is set, then a message-list specifier in the form ``/x:y'' will expand to all messages containing the substring 'y' in the header field 'x'. The string search is case insensitive. If 'x' is omitted, it will default to the ``Subject'' header field. The form ``/to:y'' is a special case, and will expand to all messages containing the substring 'y' in the ``To'', ``Cc'' or ``Bcc'' header fields. The check for ``to'' is case sensi-

tive, so that ``/To:y'' can be used to limit the search for 'y' to just the ``To:'' field.

verbose
: Setting the option verbose is the same as using the -v flag on the command line. When mail runs in verbose mode, the actual delivery of messages is displayed on the user's terminal.

Option string values

EDITOR
: Pathname of the text editor to use in the edit command and ~e escape. If not defined, /usr/bin/ex is used.

LISTER
: Pathname of the directory lister to use in the folders command. Default is /bin/ls.

MBOX
: The name of the mbox file. It can be the name of a folder. The default is ``mbox'' in the user's home directory.

PAGER
: Pathname of the program to use in the more command or when the crt variable is set. The default paginator more(1) is used if this option is not defined.

REPLYTO
: If set, will be used to initialize the Reply-To field for outgoing messages.

SHELL
: Pathname of the shell to use in the ! command and the ~! escape. A default shell is used if this option is not defined.

VISUAL
: Pathname of the text editor to use in the visual command and ~v escape. If not defined, /usr/bin/vi is used.

crt
: The valued option crt is used as a threshold to determine how long a message must be before PAGER is used to read it. If crt is set without a value, then the height of the terminal screen stored in the system is used to compute the threshold (see stty(1)).

escape
: If defined, the first character of this option gives the character to use in the place of ~ to denote escapes.

folder
: The name of the directory to use for storing folders of messages. If this name begins with a '/', mail considers it to be an absolute pathname; otherwise, the folder directory is found relative to your home directory.

indentprefix
: String used by the ~m tilde escape for indenting messages, in place of the normal tab character ('^I'). Be sure to quote the value if it contains spaces or tabs.

record
: If defined, gives the pathname of the file used to record all outgoing mail. If not defined, then outgoing mail is not so saved.

screen
: Size of window of message headers for z.

sendmail
: Pathname to an alternative mail delivery system.

toplines If defined, gives the number of lines of a message to be printed out with the top command; normally, the first five lines are printed.

ENVIRONMENT

mail utilizes the HOME, LOGNAME, USER, SHELL, DEAD, PAGER, LISTER, EDITOR, VISUAL, REPLYTO MAIL, MAILRC, and MBOX environment variables.

If the MAIL environment variable is set, its value is used as the path to the user's mail spool.

FILES

/var/mail/*	post office (unless overridden by the MAIL environment variable)
~/mbox	user's old mail
~/.mailrc	file giving initial mail commands; can be overridden by setting the MAILRC environment variable
/tmp/R*	temporary files
/usr/share/mailx/mail.*help	help files
/etc/mail.rc	system initialization file

SEE ALSO

fmt(1), newaliases(1), vacation(1), aliases(5), mailaddr(7), mail.local(8), newaliases(8), sendmail(8)

"The Mail Reference Manual", /usr/share/doc/mailx.

HISTORY

A mail command appeared in Version 3 AT&T UNIX. This man page is derived from The Mail Reference Manual originally written by Kurt Shoens.

BUGS

There are some flags that are not documented here. Most are not useful to the general user.

Usually, Mail and mailx are just links to mail, which can be confusing.

Thanks

☺ Thanks for reading this tutorial, I hope it helped; If you have any suggestions (during the Fall 08 quarter - preferably), feel free to email me them using the link under my name on the cover page.