

Slide 1

### Administrivia

- Reminder: Homework due Wednesday.
- Is that it on homework? Almost — one more homework, a bit later, asking you to reflect on the course.
- Anything else? Syllabus mentions a project. I think we still have time for a scaled-down version. Details soon.
- Minute-essay responses have been great! I haven't made time to reply but I've enjoyed reading them.
- Current (as of about 5pm Monday) attendance report uploaded. If you're not sure whether you've sent me all the minute essays ...

Slide 2

### Mail on UNIX Systems — Traditional Approach

- (Reasonably good ASCII-art picture in rather old HOWTO for administrators, recommended as reading for today.)
- MTA ("mail transport agent") communicates with the outside world to send/receive mail, using SMTP (Simple Mail Transfer Protocol). Several choices; typically choose one when installing/configuring. Traditional one is `sendmail`; very powerful, but not easy to configure. Losing ground on desktop-oriented Linux distributions to alternatives such as `postfix`.
- MDA ("mail delivery agent") delivers mail locally, often to "mail spool". One choice is `procmail`, which allows various forms of filtering.
- MUA ("mail user agent") is what users interact with directly. Many choices (more shortly).
- Mechanisms for running "batch" jobs (including `at` and `cron`) often mail output to owning user.

### Mail on UNIX (and Other) Systems — More-Current Approach

Slide 3

- MUA communicates directly with mail server (probably not local), using IMAP, POP3, etc. Mail stays on server. Requires that mail server be running something that provides IMAP/POP3/etc. access.
- Another alternative is Web-based mail client.
- Can co-exist with traditional approach.

### Mail Delivery

Slide 4

- Normally, mail gets delivered to the system "mail spool". (For `@cs.trinity.edu` addresses, on Sol.)
- To forward mail elsewhere, create a text file `.forward` in your home directory. In it put the forwarding address(es). If one of them is `\username`, one copy goes to regular mail spool. (**CAUTION:** The backslash is important to avoid recursive delivery.) Can also use this to put mail in a file.
- Can also use `procmail` to redirect/filter incoming mail.

## Reading and Sending Mail

Slide 5

- Lots of programs you can use to read mail (MUAs). Most allow reading from different sources:
  - System mail spool. (For us, mail sent to `cs.trinity.edu` goes to department mail server. You *can* log in but probably should not.)
  - Another file (e.g., file in home directory).
  - Server that provides POP3 or IMAP access.
- MUAs also, of course, allow sending mail. Some have built-in support for outgoing mail, usually (?) via SMTP. Others turn over outbound mail to system MTA (e.g., `sendmail`). MUA may need to authenticate both to read and send mail. (Why should it matter for sending mail? if not, MTA is an “open relay” and a community nuisance.)

## MUAs

Slide 6

- Several text-based MUAs: I use `mutt`; Dr. Zhang used to use `pine` and still has her students use it and successor `alpine` for submitting homework (I think?). Others include `mail` and `elm`. Many use user-specified external editor to compose mail. Can also “do” e-mail from within `emacs`.
- Also several GUI-based MUAs: Thunderbird, Evolution, Mail (OS X), etc.

## Sending Mail from the Command Line

- Simplest / most primitive program for sending (and reading) mail is `mail`. Reasonable for sending pre-composed text-only messages. Example:

```
echo "this is a test" | \  
    mail -s "test" bmassing@cs.trinity.edu
```

(\ is how you tell the shell to ignore end-of-line.)

- What about attachments? Next slide(s) ...

Slide 7

## Mail and Attachments

- Mail traditionally plain-text. How then to represent attachments?
- Old way is to use some way of encoding non-text data as plain text, which then becomes the message body. Possibilities include:
  - Encode files to attach with `shar`. Recipient pipes message body through `unshar`.
  - Encode files to attach with `uuencode`. Recipient pipes message body through `uudecode`.
- More-recent standard for mail is (includes?) MIME ("Multipurpose Internet Mail Extensions"), which provides for standard ways of including non-text in e-mail.

Slide 8

### Files for Local Mail

Slide 9

- Local mail stored in text(!) file(s). Two traditional formats:
- “mbox” represents each mailbox as flat files, messages separated by lines beginning “From ”. Older format, and what I use.
- “maildir” represents each mailbox as a directory, one (flat) file per message. Newer format and seems to make more sense.
- Can the two coexist? I don’t know! but there are tools for converting.

### Mail — Some Technical Stuff

Slide 10

- Original standard for e-mail (RFC 561?) was text-only, “7-bit ASCII-clean”, though one could encode non-text files in some way compatible with that.
- (“RFC”? Literally, “Request For Comments”, but these are generally documents describing standards for Internet-related things.)
- E-mail message consists of “envelope” (real sender), “headers” (something: something), and “body”.  
(I sometimes find it useful to look at envelope. Many MUAs don’t make that easy.)
- Newer standard is MIME (“Multipurpose Internet Mail Extensions”), which extends previous standard to allow non-ASCII text and multipart messages. Includes notion of “MIME types” (types of files).

## Sending Mail from the Command Line with (MIME) Attachments

Slide 11

- Sending mail from the command line with MIME attachments requires a program that can be run from the command line and knows how to properly include attachments with the outgoing text.
- `mutt` is one such. Example of its use:

```
echo "here are my files" | \  
  mutt -s "my files" -a file1 file2 -- \  
  bmassing@cs.trinity.edu
```

(The `--` indicates the end of the list of files to attach. `\` is how you tell the shell to ignore end-of-line.)
- (Review `mail-files` script.)

## Filtering Mail with `procmail`

Slide 12

- `procmail` can be used for many kinds of “filtering” operations on mail. Selected messages can be saved (to files), forwarded, automatically replied to, or passed to other programs.
- On many UNIX systems, you make this happen via a `.forward` file. Other systems (such as Red-Hat-based Linux) use `procmail` as the MDA, so all that’s needed is to put a file `.procmailrc` in your home directory.
- Syntax for `.procmailrc` can be intimidating, but `man` pages for `procmail`, `procmailrc`, and `procmaillex` have examples that can help.
- One use of `procmail` is to route incoming mail to a file other than the system mail spool. Can be useful if the other file is NFS-mounted (as `/users` is on our “client” machines). Another use is to run all incoming mail through a spam-filtering program, such as `spamassassin`.

## spamassassin

Slide 13

- As its name maybe suggests, `spamassassin` is a program that tries to identify and flag “spam”. Supposedly “trainable” (meaning that, in addition to whatever built-in rules it has, it can build/maintain a database that stores information about previously processed messages).

Written in Perl so not very fast, but an option (which we do on our mailserver) involves background processes (`spamd`) and a client program (`spamc`) that communicates with them.

- Installed on department mail server, so you could try it out by including a line in your `.procmailrc` file.
- How to install on your own machine depends on distribution. I found several fairly useful-sounding HOWTO documents on the Web, *but* many of them also tell you how to route all incoming mail through `spamassassin`, which you can skip if you’re willing to use `procmail` as described.

## Retrieving Mail with fetchmail

Slide 14

- `fetchmail` is designed to — fetch mail, and turn it over to local MTA for delivery. Can retrieve mail from different sources using different protocols (POP3, IMAP, etc.). Very flexible/configurable.
- Could use this on your own machine to retrieve mail from Sol (and/or other servers), or on lab machines to retrieve mail from other servers.
- (Historical/cultural aside: A widely-cited essay related to open-source software, “The Cathedral and the Bazaar” by Eric Raymond, was based in part on his experiences with `fetchmail`.)

Slide 15

### A Little About `mutt`

- Text-based mail client. Probably not very widely used but appears to have a devoted user base.
- Highly configurable, though sometimes it takes some tinkering to get things set up just so.
- By default, reads from and writes to local files and uses local MTA. But can also configure to read from POP or IMAP server and (newer versions) use remote SMTP server for outgoing mail.
- Uses an external editor to compose messages. Configurable.
- Has built-in support for showing text (“pager”) and maybe a few other MIME types. Uses external programs to display others, based on mime-types and “mailcap” configuration files.
- Online help via F1 (but that doesn’t work if intercepted by GNOME etc. — but F2 should work too).

Slide 16

### A Sample Configuration for `mutt` on Our Machines

- `.forward` to route mail to department mail server. (File should contain a single line, `\username@cs.trinity.edu`. That leading backslash avoids potential forwarding loops.)
- `.procmailrc` to put mail in file in your home directory.
- `.spamassassin/user_prefs` to set `spamassassin` preferences.
- `.muttrc` for `mutt` configuration.
- (Look at sample configuration file(s).)



### Minute Essay

- What programs have you used to read mail? What do you like/dislike about each?

Slide 17