

Slide 1

Administrivia

- Reminder: Homework 7 due Monday.
- Information about projects coming soon.

Slide 2

Minute Essay From Last Lecture

- More than one person said, reasonably enough, that since they knew little about these kinds of tools coming into the course . . .
- One person asked about differences between how (or whether) commands work depending on whether executed in a terminal window or remotely.

Just For Fun — “Extreme” ASCII Art?

- Some of you may have heard of “ASCII art”? a truly over-the-top example, from quite a while ago, can still be found, via

```
telnet towel.blinkenlights.nl
```

(to interrupt control-] then control-d).

Slide 3

- (What some people choose to do with their time can be — interesting?)

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). Choice made by sysadmin. A 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 later).

Slide 4

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

Slide 5

- MUA communicates directly with mail server (probably not local), using IMAP, POP3, etc. Mail stays on server. Examples include Thunderbird and Evolution. 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 6

- 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. Can also use this to put mail in a file.
- Can also use `procmail` to redirect/filter incoming mail.

Reading and Sending Mail

Slide 7

- 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 (“Simple Mail Transport Protocol”). Others turn over outbound mail to system MTA, e.g., `sendmail`. Not a concern on lab machines, but may be if using mail on your own machine.

Sending Mail from the Command Line

Slide 8

- Simplest / most primitive program for sending (and reading) mail is `mail`. Pretty reasonable for sending pre-composed text-only messages. Example (\ here is how you tell the shell to ignore end-of-line):

```
echo "this is a test" | \  
    mail -s "test" bmassing@cs.trinity.edu
```
- What about attachments? Next slide(s)...

Mail and Attachments

Slide 9

- Mail is 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.

Sending Mail from the Command Line with (MIME) Attachments

Slide 10

- 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.
- `pine` is one such, but it's disappearing. `mutt` is another Example of its use (`\` here is how you tell the shell to ignore end-of-line):

```
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.)

Slide 11

Filtering Mail with `procmail`

- `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 `procmailex` 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`.

Slide 12

Retrieving Mail with `fetchmail`

- `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`.)

MUAs and Other Mail-Related Programs

Slide 13

- Many text-based MUAs: I use `mutt`; Dr. Howland and Dr. Zhang used to use `pine`. others include `mail` and `elm`. Many use user-specified external editor to compose mail. Can also “do” e-mail from within `emacs`.
- GUI-based MUAs include Thunderbird, Evolution, Mail (OS X).
- Also programs to provide Web access to mail — e.g., `squirrelmail` (installed on Sol, “CS Web Mail” on department home page).

Filtering Mail with `spamassassin`

Slide 14

- 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.

Mail in UNIX — `mutt`

Slide 15

- As mentioned, one text-based mail program that's popular in some circles is `mutt`.
- Default setup reads from local mail spool, but can also be configured to read from remote mail server via POP or IMAP.
- *Highly* configurable via text configuration file(s) `.muttrc`. Can be configured to use outside programs to display various kinds of attachments and even HTML mail.
- Like many traditional-UNIX programs, makes use of an external program for text-editing (here, composing mail).

Minute Essay

Slide 16

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