The Home-Directory Mail System

Dr Andrew J Findlay

Manufacturing and Engineering Systems
Brunel University
Uxbridge
UB8 3PH

Andrew.Findlay@brunel.ac.uk

July 1988

ABSTRACT

To make the mail system behave in a more obvious way on networks of workstations, mailboxes are moved into users' home directories. User Agent programs are modified, and a new mail notification system is introduced.

1. Background

Brunel University has a large network of Unix-based computers. There are about fifty machines covering seven departmental subdomains. There are four different processor types. All machines use Sun's NFS, and all users have a single home directory that is exported to all machines where they might log in. Most workstations are in communal areas so a user is likely to log in on a different machine each day.

The mail system is based on the *UK sendmail 1.4a* package [1], and users create *.forward* files to specify which machine their mail should be delivered to. This approach worked well when there were only a few machines, but has become increasingly awkward as the network has grown. To avoid locked-up machines, most user filesystems are soft-mounted. This means that there can be times when a machine is running but a group of user directories are not present. If mail arrives at such a time, the *.forward* file is not effective. A user's mail could be spread around the whole network, with one or two messages on each machine.

With increasing use being made of the mail system, it was felt that some improvements were needed.

2. Options

- Use a big alias file.
  This would avoid the 'invisible forward file' problem, but it still requires each user to nominate *one* machine where they will read mail.
- Mount a single /usr/spool/mail directory for the whole Campus.
  This is probably the simplest solution, but there would be problems with mail delivery agents that run as root. For security reasons, NFS maps *root* into *nobody* when accessing remote filesystems. Mailers that expect to use lockfiles or *flock(2)* would have problems too.
- Mount one /usr/spool/mail directory for each subdomain.
  It would be possible to arrange that only the (NFS) server machine ever delivered mail. This would remove the root-mapping problem, but would not solve the lockfile problem. Many users are registered in more than one subdomain, so this solution would not help them.
- Create a Campus-wide 'mail server' as a TCP service.
  This is possibly the best solution, as it would be able to serve non-NFS and even non-Unix machines. The work involved is considerable, as all the User Agent (UA) programs currently in use expect mailboxes to be *files* in v7 mbox format.

- Move the mailbox out of /usr/spool/mail into the user's home directory.
  Although the UAs still need modifications, it was felt that this would be the best compromise between effort expended and results obtained.

3. Mailbox in Home Directory

The main benefits of this approach are:

- The system behaves in an 'obvious' way.
  If the user's home directory is available, then so is their mailbox.
- Mailboxes become subject to the same quota limits as other files.
- The existing mailbox format can be retained.

There are some problems remaining:

- Mail User Agent programs need modifying to reflect the new mailbox position.
- Some UAs react badly to the disappearance of a soft-mounted filesystem.

3.1. Files and Lockfiles

The v7/Bsd convention places all mailboxes in /usr/spool/mail and names them with the username of their owner. Lockfiles are named by appending '.lock' to the mailbox name.

The home-directory mail system names all mailboxes '.mailbox' and all lockfiles '.mailbox.lock'. Mailboxes are identified by the directory they are placed in. The /usr/spool/mail area is not used at all.

Mailbox locking is the least satisfactory part of the home-directory mail system. Sun NFS is stateless; this means that the conventional 'exclusive create' mechanism cannot be trusted. None of the traditional atomic file operations can be used, as NFS does not guarantee atomicity. The 'correct' way to lock files is to use *flock,* which makes use of the lock daemon to handle local and remote files in the same manner. Unfortunately, not all implementations of NFS support the lock daemon. As a compromise, it was decided that home-directory mail would use lockfiles *and flock.* This would give complete safety between machines supporting *flock* and reasonable safety on all others.

The mail delivery agent may have to create a new mailbox to deliver into. In this case, the lockfile must be created before the mailbox. The *flock*-style lock is therefore always applied to the lockfile.

3.2. Delivery

A new program called *hdmail* was written to handle mail delivery. It is simplified by not doubling as a User Agent. *hdmail* is aware of the peculiarities of soft-mounted filesystems: if a directory is missing or a file suddenly becomes inaccessible, the value EX_TEMPFAIL is returned to *sendmail.* Mail for users whose home directories are not available will be held in the mail queue, and delivery will be attempted again at each queue run. This has the useful side-effect that *sendmail* will look for the *.forward* file again at each queue run, making the forwarding mechanism much more reliable. If the necessary directory does not appear within a reasonable time, *sendmail* will return the message to the sender.

hdmail runs as the recipient user, not as root. This gets round the root-mapping problem on remote filesystems, and also improves security. When secure NFS [2] is introduced, it may be necessary to have mail delivered only by the machine that supplies the user's home directory.

After the mail has been delivered, *hdmail* arranges to notify the user of its arrival.

4. Notification

The home-directory mail system allows a user to read and send mail when logged in on any of the fifty-plus Unix machines on Campus. When new mail arrives, the user expects to be told about it.

Under the Bsd version of the mail system, this function is handled by the *biff* daemon. *Biff* is a UDP (datagram) service; it is sent a packet containing a username and a mailbox-offset. If the named user is logged on, the mailbox is inspected from the given offset to construct a suitable message for display on their terminal.

Biff provides a good service on individual computers. It is not so useful on a large network of workstations, as a user may not be logged in on the machine that delivers the mail.

Four approaches to the notification problem were considered:

(1) Modify *biff* to read the new mailboxes, and invoke it on all known machines.

(2) Use a background 'watcher' process, such as *newmail* from the *elm* suite.

(3) Use the shell's MAIL variable.

(4) Create a new network service.

With the number of workstations and the volume of mail both increasing, it was felt that (1) would soon impose an excessive load on the system.

Solution (2) was ruled out as it would almost double the number of active processes on undergraduate teaching machines.

Solution (3) would be a backward step, as users would only find out about new mail when they returned to the shell.

It was decided to create a new network service, similar to *biff,* but with less work done by the daemon. The new service, *tell,* is a UDP service that receives a packet containing a username and a message. If the named user is logged in and has set the *'biff'* bit on their tty, the message is written to that terminal. For security reasons, only a limited set of control characters is allowed in the message. Any 'illegal' characters are converted to spaces by the daemon. The *tell* daemon provides a more general service than *biff.* It is not limited to mail notification, though that is its only function at the moment.

To avoid having to invoke *tell* on every machine on the network, another file is maintained in the user's home directory. The '.whereami' file records the name of each machine where the user has logged in, together with the time of login. When *hdmail* is ready to notify the user that new mail has arrived, it scans '.whereami' and sends a *tell* packet to every machine where the user has logged in within the last 48 hours. If the file does not exist or if no recent logins are recorded, a *tell* packet is sent to 'localhost'.

The '.whereami' file is maintained by a program called *mailcheck* that is invoked at login time from '.login' or '.profile'. *mailcheck* also prints a message if mail is waiting to be read, thus avoiding the need to modify the shells and the login program.

5. User Agents

The major UAs in use at Brunel are *elm, Mail* (/usr/ucb/mail), and *mailtool* (under SunWindows only). Since *mailtool* is a driver for *Mail,* it was only necessary to modify two UAs to satisfy almost all the users.

elm was tackled first, taking version 1.5b as the starting point. References to mailboxes are scattered in the source code, which made the conversion rather tricky. The utility programs had to be modified individually. *elm* still uses lockfiles only; it is hoped that version 1.7 or 2.0 will have a cleaner interface to the mailbox, allowing the full locking protocol to be used.

Mail was modified using source code from the 4.3Bsd distribution. The changes required were confined to two source files. Unfortunately, the 4.3 version of Mail is not compatible with *mailtool* so this work had to be repeated when Sun source code was obtained.

Other UA programs have been converted, including *prmail,* but not */bin/mail* (which is superceeded by */usr/ucb/mail*). The interface between mail and *emacs* has not yet been converted. Other minor links to the mail system (such as the News '(Mail)' prompt) can safely be ignored.

6. Mail Transport Agents

Brunel uses *sendmail* for transport and routing of mail. The only modifications needed were in the part of the configuration file describing the 'local' mailer. *hdmail* replaces */bin/mail* for local mail delivery. The only major difference is that *hdmail* can only deliver to one user per invocation.

7. Operating Experience

Nobody noticed.... Apart from one incident when an automatic updating system installed the modified UAs on a machine that was not due for conversion, the transition has been smooth. Problems were encountered when re-compiling *Mail* on certain machine types: in one case caused by an optimiser bug, and in the other by bad memory-allocation code.

The home-directory mail system is now operating in three of Brunel's six Unix-based subdomains, on two different CPU types. It is to be installed in the remaining subdomains shortly, which will include two more CPU types.

References

1 Crammond, J., "Domain Mailers: Sendmail", Workshop on Networking, UKUUG, December 1987

2 Taylor, B., "A framework for Network Security", Sun Technology, Spring 1988