Toolman: Sorting and Archiving Email

In this article, I'll discuss a methodology for sorting email into mailboxes based on year and month, which can then be compressed for archival purposes. In addition, I'll cover retrieval techniques, and I'll survey some related tools.

by Daniel E. Singer

Dan has been doing a mix of programming and system administration since 1983. He is currently a system administrator in the Duke University Department of Computer Science in Durham, North Carolina, USA.

<des@cs.duke.edu>

Tools: *sortmail, decomposemail, recomposemail*

Abstract: *sort email messages by date/time into monthly mailboxes*

Platforms: *most UNIX*

Language: *Bourne shell*

Author: *Daniel E. Singer* <des@cs.duke.edu>

Availability: <http://www.cs.duke.edu/~des/toolman.html> <ftp://ftp.cs.duke.edu/pub/des/scripts>

Hoarding

If you're like me, you're a pack rat with your email: you stow it away somewhere, but never like to get rid of it or take it offline. After all, you never know when you're going to need to `grep` through it to find some vital instructions, reconstruct a conversation, or verify that you or someone said something 27 months ago. All this old email takes up a lot of disk space. And some of us live within quotas. (I'm currently struggling to stay within a 100MB quota, more on principle than necessity.)

So what's an email hoarder to do? Some people use any of various mail filters to automatically sort incoming email into mailboxes (sometimes known as folders) and even discard certain messages as they come in (can you say "spam"?). Examples are `procmail` [1, 2] and the bundled filtering features of `elm` [3]. But I'm kind of old-fashioned and distrustful of these filters: I like to decide on a case-by-case basis which messages to put where, and how long they should hang around in my *inbox*, saving or deleting them as seems appropriate. What tends to happen is messages pile up in my *inbox*, and periodically I'll go through and save some old messages to mailboxes and purge out others. As I do this, messages get saved out of chronological order – sometimes *very* out of order. This may or may not resemble your email processing practices.

To complete this picture, let me add that I save messages to mailboxes using filenames based on the username of the sender or the name of a company, product, or concept, along with certain upper- and lowercase conventions. (Occasionally, I'll even save a message to more than one mailbox because the concepts of links and cross-posting are not available in this context.)

Sorting and Chunking

What I want is a way to save my email, archive it in manageable chunks, compress it, and still keep it useful. (Yes, I want to have my cake and eat it!) I could just periodically move mailboxes to an archive directory, add sequence numbers to the filenames, and compress them; but each such archive would not necessarily be sequential over some period, and searching would be more difficult than it could be. I want to be able to search through email by time periods as well as by some person or topic. Also, some mailboxes tend to get very large and unwieldy (that is, slow), so splitting them into chunks should also be a performance gain.

The methodology I've come up with for this is to disassemble mailboxes into their component messages, sequence them by date/time, and then reassemble them into mailboxes by year/month, optionally storing these into monthly subdirectories. For instance, I have a mailbox named "USENIX," and since it tends to collect a lot of messages, I occasionally want to *chunk* it (not *chuck* it!). I can do this by going to my mail directory, and running the `sortmail` script.

```
% cd ~/mail
% sortmail -mc USENIX
```

This will create (or append to) mailboxes with names like "USENIX.9805" for May of 1998, "USENIX.9806" for June of 1998, and so on. Each such mailbox will hold the messages for that month of that year only, sorted meticulously by date and time. Alternatively, we could have used `-M` instead of `-m`, telling `sortmail` to instead deposit

the sorted email into monthly subdirectories, yielding mailboxes such as "9805/USENIX," "9806/USENIX," etc. In either case, `sortmail` will append to the monthly files if they already exist. And if any such monthly mailboxes are already compressed (via `compress` or `gzip`), `sortmail` will first decompress them, then add the new messages, and then recompress them. If the `-R` (recurse) flag is used, any appended mailboxes will also be resorted. The `-c` flag tells `sortmail` to move any messages for the current month back to the mailbox of the original name, in this case "USENIX."

I tend to prefer the YYMM/mbox scheme over the mbox.YYMM one, because I currently have around 500 mailboxes in my mail directory, and the latter scheme would add too much additional clutter.

Another tool similar to `sortmail` is similarly named `mailsort`. It is written in Perl by Andras Salamon (<http://www.dns.net/andras/>). `mailsort` can also reverse sort and is styled after the UNIX filter model much more so than `sortmail`. It is fast and robust, though it lacks the monthly chunking features of `sortmail`. You can pick up `mailsort` at your fave CPAN[4] site under <.../scripts/mailstuff/mailsort.tar.gz>.

Safeguarding

To safeguard your precious data (and mine), `sortmail`, by default, will also: (1) create a subdirectory into which it backs up any mailboxes that it is going to change and (2) create an additional subdirectory into which it copies mailboxes to be sorted and in which it does all of its work. So if you're a little nervous about letting this stuff loose on your mailboxes, everything is covered. (Of course, you can make additional copies or copy the mailboxes to another directory and do it there until you get the feel of it.) After running `sortmail`, you can verify that things are OK, and then remove the backup copies. Then you can compress any of the older mailboxes if desired.

Supporting Cast

The `sortmail` script is a higher level interface to two scripts that do a lot of the work: `decomposemail` and `recomposemail`. Their names are indicative of their functions: the first breaks up a mailbox into files, each containing an individual message; the second reassembles the messages in sorted order. They each can be used standalone, though `sortmail` saves many manual steps and does add additional functionality such as making backups, working in a subdirectory, appending to existent files, and recursing.

Searching

Now, let's say you've been using `sortmail`, and you have subdirectories such as "9601," "9602," ..., "9807." Furthermore, you have already compressed all the mailboxes in the subdirectories for the months in 1996 and 1997. Now you want to find that cornbread recipe that your mom emailed you a year or two ago (and you don't feel like calling). Well, you don't want to go and uncompress all those files, and you probably don't want to type a bunch of awkward commands like:

```
% gzcat 9701/mom | grep -i cornbread
```

A tool that you can use for this sort of situation is `grepz`. It will uncompress on the fly (without modifying your files) and can even recurse through a directory hierarchy if given half the chance. So the line

```
% grepz -i cornbread 9[67]??/mom
```

would do the trick. In the event that you didn't know who sent you that recipe or when, a bigger hammer would be

```
% grepz -i cornbread .
```

What I want is a way to save my email, archive it in manageable chunks, compress it, and still keep it useful.

More Tools:	*grepz; rotatemail; check*
Abstract:	*search for patterns; rotate files monthly; maintain index files*
Platforms:	*most UNIX*
Language:	*Bourne shell*
Author:	*Daniel E. Singer* <des@cs.duke.edu>
Availability:	<ftp://ftp.cs.duke.edu/pub/des/scripts>

If you've got too many mailboxes and other files and subdirectories under your mail directory, another problem can be just keeping track of what's what.

This would search through all files and subdirectories recursively. `grepz` will also handle noncompressed files properly.

A similar search tool that can handle compressed files is a Bourne shell script named `zgrep` that comes with the `gzip`[5] utility archive. Another very handy search tool, written in Perl by Jeffrey Haemer and Jeffrey Copeland, is named `mgrep`[6] and is designed specifically for searching mailboxes. It returns entire messages that are matched, instead of just matched lines. These could be combined with `find` and `xargs` to approximate the recursive behavior of `grepz`.

```
% find . -name occult\* -print | xargs mgrep -i voodoo | less
```

A more generalized approach for dealing with compressed files is the `zloop` shell script by Jerry Peek. You can tell it to run the command of your choice on a group of compressed files. `zloop` is discussed in the book *UNIX Power Tools*[7].

```
% zloop 'mygrep -3d "on the road"' outbox.*.gz
```

Rotating

Another script that operates in this scheme of things is called `rotatemail`. I use it at the start of each month via UNIX's `cron` utility to automatically rename my "outbox" file congruent with `sortmail`'s monthly naming scheme. Sorting isn't necessary here since outboxes tend to be sorted already. A `crontab` entry like

```
0 0 1 * * rotatemail /home/you/mail/outbox 2>&1
```

will rename your outbox to "outbox.9807," assuming that July 1998 just ended. It will then create a new, empty outbox file with appropriate permissions. If you prefer the monthly subdirectory scheme, yielding a filename like "9807/outbox," then just add the `-M` flag. Of course, this could be used on files other than just your outbox.

Tracking

If you've got too many mailboxes and other files and subdirectories under your mail directory, another problem can be just keeping track of what's what. I've recently started using `check`[8] to create and maintain an INDEX file in my mail directory. This helps me to have a short description of each mailbox, to group them into categories, and to isolate duplicates that can be combined and junk that can be deleted. You might find this useful as an additional means of riding herd on your mailboxes. Then again, there's always that memory enhancement course you've been meaning to take!

Ending

A lot of territory has been covered here. My hope is that you can mix and match these tools and techniques to suit your taste. You might even want to add a few of your own design.

If you find that any of my tools don't work properly on your UNIX platform, drop me a line, and I'll pound on them for you. Just ask Bruce Foster at Northwestern University (<http://charlotte.acns.nwu.edu/bef/>). I recently fixed `seepath`[9] to work in his HP-UX/DFS/Posix-shell[10] environment!

I have a few other scripts that deal with mailbox manipulations. I'll leave them at the FTP location in case you're interested. As usual, please let me know if you have any comments or suggestions.

Notes

[1] `procmail` is written by Stephen R. van den Berg (<berg@pool.informatik.rwth-aachen.de>) at RWTH-Aachen, Germany, <ftp://ftp.informatik.rwth-aachen.de/pub/packages/procmail/procmail.tar.gz>.

[2] An interesting article about `procmail` and email filtering in general is Jeffrey Copeland and Jeffrey S. Haemer. Work: Not Looking Through Our Mail. *SunExpert Magazine*, May 1998, pp. 72-75. <http://www.alumni.caltech.edu/~copeland/work.html>.

[3] `elm` is maintained by the Elm Development Group, <http://www.myxa.com/elm.html>.

[4] Comprehensive Perl Archive Network. See <http://www.perl.org/> for the site nearest you.

[5] `gzip` is maintained by the Free Software Foundation. See <http://www.fsf.org/order/ftp.html> for the site nearest you.

[6] Jeffrey Copeland and Jeffrey S. Haemer. Work: Looking Through Our Mail. *SunExpert Magazine*, March 1998, pp. 80–84. <http://www.alumni.caltech.edu/~copeland/work.html>.

[7] Jerry Peek, Tim O'Reilly, and Mike Loukides. *UNIX Power Tools*, 2nd ed. Sebastopol, CA: O'Reilly & Associates, 1997. <ftp://ftp.ora.com/pub/examples/power_tools/unix/split/zloop>.

[8] Daniel E. Singer. ToolMan's Approach to Documenting UNIX Directories. *;login:*, June 1997, pp. 45–48.

[9] Daniel E. Singer. ToolMan: Upcoming Tools; Analyzing Paths. *;login:*, February 1998, pp.40–43.

[10] The Bourne shell drops implicit null arguments when parsing a string into positional parameters. The Posix-compliant shell on HP-UX (and possibly other platforms) does not, and this was causing `seepath` to choke.

Got a tool that's useful, unique, way cool? Please send a description to <Toolman@usenix.org>.