extsmail

Laurence Tratt
http://tratt.net/laurie/

Bournemouth University

2008/11/19

Overview

1. Motivation.
2. Idea.
3. Options.
4. Path chosen.
5. Implementation.
6. Thoughts.

Motivation

- How to send e-mail?

Motivation

- How to send e-mail?
- Traditional solution: SMTP.
- Problems: when moving between networks; authenticating; servers tend to be in someone elses control; unreliable networks.

Motivation

- How to send e-mail?
- Traditional solution: SMTP.
- Problems: when moving between networks; authenticating; servers tend to be in someone elses control; unreliable networks.
- Restate: how to send e-mail when moving around (often unreliable) networks?

Motivation

- How to send e-mail?
- Traditional solution: SMTP.
- Problems: when moving between networks; authenticating; servers tend to be in someone elses control; unreliable networks.
- Restate: how to send e-mail when moving around (often unreliable) networks?
- Alternative: local sendmail.
- Problem: blacklisting on dynamic IPs.

Motivation

- How to send e-mail?
- Traditional solution: SMTP.
- Problems: when moving between networks; authenticating; servers tend to be in someone elses control; unreliable networks.
- Restate: how to send e-mail when moving around (often unreliable) networks?
- Alternative: local sendmail.
- Problem: blacklisting on dynamic IPs.
- Alternative*: sendmail via ssh.

```
/usr/bin/ssh -q -C -l ltratt 137.73.8.80
/usr/sbin/sendmail -oem -oi
```

- Problems: unreliable networks, unreliable servers.

Idea

- A sendmail lookalike frontend to ensure compatibility.
- A backend daemon which repeatedly retries sending e-mail until it's sure its sent.

Idea

- A sendmail lookalike frontend to ensure compatibility.
- A backend daemon which repeatedly retries sending e-mail until it's sure its sent.
- Not restricted to ssh: generalise to 'external commands' *externals*.
- Allow multiple externals to be defined; try them in order.
- Allow messages to be routed to different externals depending on message contents.

Idea

- A sendmail lookalike frontend to ensure compatibility.
- A backend daemon which repeatedly retries sending e-mail until it's sure its sent.
- Not restricted to ssh: generalise to 'external commands' *externals*.
- Allow multiple externals to be defined; try them in order.
- Allow messages to be routed to different externals depending on message contents.
- And thus `extsmail` is born.

Options

- Python.
- Pros: quick (to write) and reliable (to run).
- Cons: non-standard dependency; relatively resource heavy.

Options

- Python.
- Pros: quick (to write) and reliable (to run).
- Cons: non-standard dependency; relatively resource heavy.
- C.
- Pros: available everywhere, light-weight at run-time.
- Cons: slow (to write) and oft-questioned reliability (buffer overruns, stack smashes etc.).

Path taken

- Choice: traditional, old fart C.
- Only use what comes with a ‘typical’ ‘reasonable’ Unix. e.g. lex, yacc, regex.
- Allow delivery to happen in batch or daemon mode.
- Example config:

```
group {
    external mymachine {
        sendmail = "/usr/bin/ssh -n -q -C -l user
          mymachine.net /usr/sbin/sendmail"
    }

    external bk {
        sendmail = "/usr/bin/ssh -n -q -C -l user
          bk.mymachine.net /usr/sbin/sendmail"
    }
}
```

Implementation overview

- `extsmail`: sendmail lookalike. Reads in argv and messages from stdin and dumps them in a spool file.
- `extsmaild`: delivery program. Reads spooled messages and tries to deliver them via externals.
- `extsmaild` defaults to batch mode operation. `extsmaild -d` runs as a daemon, monitoring spool directory for changes.
- `extsmail` ˜150 LoC; `extsmaild` ˜1000 LoC. All of extsmail (commmon libs, parsing etc.) ˜2000 LoC.

Implementation notes

- `lex` / `yacc` interfaces are terrible (having two parsers in one binary is a hack). Memory tends to leak. But surprisingly easy to use.

Implementation notes

- `lex` / `yacc` interfaces are terrible (having two parsers in one binary is a hack). Memory tends to leak. But surprisingly easy to use.
- File locking (between `extsmail` and `extsmaild`) is silly. 3 separate function calls that do the same job, but still can't atomically open a temporary file with `mkstemp` and gain lock.

Implementation notes

- `lex` / `yacc` interfaces are terrible (having two parsers in one binary is a hack). Memory tends to leak. But surprisingly easy to use.
- File locking (between `extsmail` and `extsmaild`) is silly. 3 separate function calls that do the same job, but still can't atomically open a temporary file with `mkstemp` and gain lock.
- File APIs are terrible. Dichotomy between file descriptors and `FILE` objects.

Implementation notes

- `lex` / `yacc` interfaces are terrible (having two parsers in one binary is a hack). Memory tends to leak. But surprisingly easy to use.
- File locking (between `extsmail` and `extsmaild`) is silly. 3 separate function calls that do the same job, but still can't atomically open a temporary file with `mkstemp` and gain lock.
- File APIs are terrible. Dichotomy between file descriptors and `FILE` objects.
- In Unix executing a command is odd: `fork` (clone process) then `exec` (replace process). But fairly easy...
- ...unlike communicating with sub-processes. `extsmaild` execs an external then pipes data to it; needs to read `stderr` in case the child dies. Deadlock is likely result. Use `poll` to help asynchronously write to `stdin` and read from `stderr`. Tricky.

Final thoughts

- Easier than I expected in some ways (despite minimalistic shackles), except pipes (deadlock hurts).
- Coding to handle all errors is fun but time consuming.
- Extra features (e.g. routing) add value over other approaches.

Final thoughts

- Easier than I expected in some ways (despite minimalistic shackles), except pipes (deadlock hurts).
- Coding to handle all errors is fun but time consuming.
- Extra features (e.g. routing) add value over other approaches.
- If you have an idea, program it!

`http://tratt.net/laurie/src/extsmail/`