

誰にでも使えるメール講座  
第3回  
「送ったメールは必ずコピー」

安岡孝一

+-----+  
! Koichi Yasuoka !  
+-----+

```
set record=ファイル名
    送したメールのコピーを、指定されたファイルに追加するようにする。
```

```
yasuoka : いきなり枠囲みですね。
root : 前回の約束だからね。うまくいったかい？
yasuoka : はい、ちゃんと~/mbox に追加されてました。

~% cat .mailrc (ぼこ)
set folder=/home/yasuoka/Mail
set record=/home/yasuoka/mbox
~% Mail -f (ぼこ)
Mail Version 9/4/88 Type ? for help.
"/home/yasuoka/mbox": 5 messages
> 1 yasuoka Sat Mar 24 17:22 13/230
  2 yasuoka Sat Mar 24 18:46 10/226
  3 yasuoka Sat Mar 24 18:46 10/226
  4 takahash Sat Mar 24 19:15 13/318
  5 matukawa Sun Mar 25 13:15 14/331
```

```
& p (ぼこ)
Message 1:
From yasuoka Sat Mar 24 17:22:52 1990
To: ochi
Status: RO
```

```
Grasshopper
  Creme de cacao (blanc) 1/3
  Creme de menthe (vert) 1/3
  Fresh cream 1/3
de shake shimasu.
```

```
& p+ (ぼこ)
Message 2:
From yasuoka Sat Mar 24 18:46:33 1990
To: takahash
Status: RO
```

```
Umeda ni "Kawaii Suizokukan" tte iu noga arukedo dou?
Hankyu no Chayamachiguchi dete nishigawa.
```

+-----+  
! Koichi Yasuoka !  
+-----+

& ■

root さんに見せたくって、そのままとっといたんです。あ、s はしましたけどね。ところで root さん。

```
root : 何だい？
yasuoka : 2人の相手に同じ内容のメールを送るには、どうしたらいいんですか？
root : アドレスを2つ書く。
yasuoka : え？ アドレスって2つでも3つでも書けるんですか？
root : もちろん。今、送りたいメールがあるならやってみようか？
yasuoka : えっと
```

```
& p5 (ぼこ)
Message 5:
From matukawa Sun Mar 25 13:15:34 1990
Received: by ginkaku.kudpc.kyoto-u.ac.jp (1.1/ku2.22)
        id AA00922; Sun, 25 Mar 90 13:15:33 JST
Date: Sun, 25 Mar 90 13:15:33 JST
From: matukawa
Message-Id: <9003250415.AA00922@kudpc.kyoto-u.ac.jp>
To: takahash, yasuoka
```

Status: RO

"Kawaii Suizokukan" de iidesuyo.

Boku mo OK.

Matsukawa

& ■

matukawa さんと takahash さんに返事を書きたいんですけど。

root : じゃあ、m matukawa takahash で...、いや、このメールの2人に返事を書くなら r5 でいいな。

m	アドレス	標準入力から入力された文字列を、指定されたアドレスにメールとして送る。アドレスは複数書いてもよい。
r	番号	標準入力から入力された文字列を、指定されたメッセージの送り主と送り先の全員にメールとして送る。
R	番号	標準入力から入力された文字列を、指定されたメッセージの送り主にメールとして送る。

番号については p に同じ。番号が省略された場合は、. が指定されたものとみなす。なお、標準入力からの文字列中、~ から始まる行は特殊な意味を持つ。主なものは以下の通り。

~p	メールの内容を表示する。
~m番号	指定されたメッセージをメールに追加する。
~r ファイル名	ファイルの内容をメールに追加する。
~h	メールのヘッダ部をエディットする。
~v	set VISUAL で指定されたエディタを起動する。
~!文字列	文字列を Unix コマンドとみなして実行する。

root : マシンによっては r と R が逆なのもあるから、注意するようにね。

yasuoka : はい。

& r5 (ぼこ)

To: takahash matukawa

■  
yasuoka : 今、どういう状態なんですか？

root : キーボードからメールをそのまま打ち込めばいいんだよ。

yasuoka : 打ち込むって？

root : 今、そこで打った内容が、そのままメールとして takahash さんと matukawa さんに送られる、ってことだ。

yasuoka : っていうことは

Ja, Kawaii Suizokukan ni 18:00 tte kotode. (ぼこ)

■  
これでいいんですか？

root : うん。~p してみてください。

yasuoka : はい。

~p (ぼこ)

-----

Message contains:

To: takahash matukawa

Ja, Kawaii Suizokukan ni 18:00 tte kotode.

(continue)

■  
これが送られるメールの内容ですか？

root : そうだよ。

yasuoka : この後にサインを追加したいんですけど。

root : そこにそのまま続けて打てば追加されるよ。

yasuoka : いえ、あの、~/ .sign ってファイルにサインが入ってるんで、それを追加したいんですけど。

root : 今ホームディレクトリにいたっけ？

yasuoka : ええ、確かそうです。

root : じゃ、~r .sign だ。~r ~/ .sign としたいところだけど、メールコマンドの中ではホームディレクトリを~で表せるとは限らないからね。

yasuoka : ~r .sign ですね。

~r .sign (ぼこ)

".sign" 3/66

■

これで読み込まれたんですか？

root : ~p してみてください。

yasuoka : はい。

~p (ぼこ)

-----

Message contains:

To: takahash matukawa

Ja, Kawaii Suizokukan ni 18:00 tte kotode.

+-----+

! Koichi Yasuoka !

+-----+

(continue)

■

途中で打ち間違いに気付いた時は、どうすればいいんですか？

root : メールの内容は~v でエディタを起動して書き直す。ヘッダ部は~h だ。

yasuoka : ヘッダ部って？

root : そうだな、ちょっとためしてみようか。~h してみてください。

yasuoka : はい。

~h (ぼこ)

To: takahash matukawa■

送り先が出ましたよ。

root : もし書き直すのなら、バックスペースで戻って打ち直せる。書き直さないなら、そのままリターンだ。

yasuoka : じゃ、そのままリターンっと。

To: takahash matukawa (ぼこ)

Subject: ■

今度は Subject: ってのが出ましたけど。

root : そこにはメールの内容の要約を 1 行以内で書く。書く必要がないなら、そのままリターン。

yasuoka : そのままリターン。

Subject: (ぼこ)

Cc: ■

これは？

root : カーボンコピー。このメールのコピーを送りたい人を書く。

yasuoka : To: とはどう違うんですか？

root : To: に書かれてる人と同じように Cc: のアドレスにも同じメールが送られるから、別に違いはないんだけど、まあ送りたい相手なのか単なるおまけなのかの気持ちを表すために使うことが多いね。

yasuoka : ふーん。

Cc: (ぼこ)

Bcc: ■

今度のは？

root : ブラインドカーボンコピー。カーボンコピーと同じように Bcc: のアドレスにもメールが送られるんだけど、それが To: の人や Cc: の人にはわからないようにしてくれる。

yasuoka : うーん、色々書けるんですね。

Bcc: (ぼこ)

(continue)

■

これで終わりですか？

root : うん、~h はこれで終わり。ヘッダ部に書けるのは、この 4 つだ。

yasuoka : じゃ、次は~v の方を教えて下さい。

root : うーん、set VISUAL してないから、ちょっとまずいな。あとでプロンプトに戻った時に set VISUAL=vi とでもして、またメールを送る時にためしてみてください。

set VISUAL=コマンド メール送信中に~v で起動するエディタを指定する。
--

yasuoka : わかりました。じゃ、このメール、もう送ろうと思うんですけど、どうすればいいんですか？

root : エンドオブファイル、つまりコントロール D だ。

yasuoka : コントロール D ですね。

EOT

& ■

あ、プロンプトに戻った。これで送られたんですか？

root : うん。fi & で mbox を読み直してごらん。

yasuoka : はい。

& fi & (ぼこ)

```
"/home/yasuoka/mbox": 6 messages 1 new
  1 yasuoka      Sat Mar 24 17:22  13/230
  2 yasuoka      Sat Mar 24 18:46  10/226
  3 yasuoka      Sat Mar 24 18:46  10/226
  4 takahash     Sat Mar 24 19:15  13/318
  5 matukawa     Sun Mar 25 13:15  14/331
>N 6 yasuoka     Sun Mar 25 15:00   8/171
```

& ■

増えてる。

root : set record=/home/yasuoka/mbox してあるからね。

yasuoka : そうでしたね。

& p (ぼこ)

```
Message 6:
From yasuoka Sun Mar 25 15:00:56 1990
To: matukawa takahash
Status: R
```

Ja, Kawaii Suizokukan ni 18:00 tte kotode.

```
+-----+
! Koichi Yasuoka !
+-----+
```

& ■

いつもはメールを送るのに「Mail アドレス < ファイル名」ってしてるんですけど、あれはどういう意味なんですか？

root : 「Mail アドレス」を実行すると、メールコマンドの中で「m アドレス」を実行したのと同じ状態になるんだよ。で、標準入力からメールの内容を受けとって、エンドオブファイルでメールを送る。ただし「Mail アドレス」はメールを送り終わったら、メールコマンドを即座に終了する。

yasuoka : そういうことだったんですか。それで納得しました。

& set VISUAL=vi (ぼこ)

& ■

set VISUAL=vi も .mailrc に書いておいた方がいいんですか？

root : うん。

yasuoka : 他に .mailrc に書いておくといいいことってありますか？

root : たくさんあるけど...。まずはメールを送る時に便利なものからいこうか。

```
a 文字列 アドレス
   文字列をアドレスの別名とする。アドレスは複数書いてもよい。
a
   別名を全て出力する。
```

root : 例えば a taka takahash としておくと、送り先のアドレスが taka だったら takahash だとみなされる。長いアドレスをいちいち書きたくない時とかに便利だね。あるいは a tm takahash matukawa とかしておけば、アドレスに tm と書くだけでこの 2 人にメールが送られる。

yasuoka : なかなか便利ですね。そういえばいつも間違うんですけど

& ls (ぼこ)

Unknown command: "ls"

& ■

としたら h が実行されるようにはできないんですか？

root : それはできない。できるのはアドレスの別名だけ。

yasuoka : わかりました。

```
set ask
   メール送信時に Subject: を尋ねる。(一般に BSD)
set asksub
   メール送信時に Subject: を尋ねる。(一般に System V)
set askcc
   メール送信時に Cc: を尋ねる。
```

yasuoka : これ、便利なんですか？ うっとうしいだけみたいなのがするんですけど。

root : うん。ただマシンによってはこれがデフォルトで設定されてる時もあるから、うっとうしいときには .mailrc で unset を使って解除しておくんだ。

```
unset 変数
      変数をセットされていない状態に戻す。
set
      セットされている全ての変数の情報を表示する。
```

yasuoka : わかりました。  
root : 次はメールを読む時に便利な設定を教えよう。人からもらったメールを1つpしてくれるかい?  
yasuoka : はい。

```
& p4 (ぼこ)
Message 4:
From takahash Sat Mar 24 19:15:09 1990
Received: by ginkaku.kudpc.kyoto-u.ac.jp (1.1/ku2.22)
      id AA00415; Sat, 24 Mar 90 19:15:08 JST
Date: Sat, 24 Mar 90 19:15:08 JST
From: takahash
Message-Id: <9003241015.AA00415@kudpc.kyoto-u.ac.jp>
To: yasuoka, matukawa
Status: RO
```

"Kawaii Suizokukan" de iidesuyo.

takahashi

& ■

root : それじゃ ig message-id を実行してから、もう一度 p して。

```
ig 文字列
      p を実行する際、ヘッダ部のうち文字列で始まる行を表示しない。文字列は複数書いてもよい。文字列が省略された場合は、現在設定されている文字列を全て表示する。
```

yasuoka : ig message-id してから p ですね。

```
& ig message-id (ぼこ)
& p (ぼこ)
```

```
Message 4:
From takahash Sat Mar 24 19:15:09 1990
Received: by ginkaku.kudpc.kyoto-u.ac.jp (1.1/ku2.22)
      id AA00415; Sat, 24 Mar 90 19:15:08 JST
Date: Sat, 24 Mar 90 19:15:08 JST
From: takahash
To: yasuoka, matukawa
Status: RO
```

"Kawaii Suizokukan" de iidesuyo.

takahashi

& ■

あ、Message-Id: の行が表示されなくなりました。

root : さらに ig received date from status するとおもしろいんだけど、もう時間がないから後で自分でやってみてね。最後は長いメールを1ページ分ずつ表示する方法だ。

```
set crt=数 PAGER=コマンド
      「数」行を越えるメッセージを表示する際に、指定された Unix コマンドを用いる。
```

root : BSD なら PAGER=more、System V なら PAGER=pg という設定が、まあ標準かな。

yasuoka : その more とか pg とかって何ですか?

root : ファイルとかを1ページ分ずつ表示するコマンドだよ。more はスペースで、pg はリターンで次のページに進む。使ったことないのかい?

yasuoka : ありません。

root : じゃ、後で使ってみるように。さて、これでメールコマンドに関する話は終わりだ。ちょっと駆け足だったけど、理解できたかい?

yasuoka : はい、頑張って覚えていきます。どうもありがとうございました。