



よしだともこの Linux 事始めの書

第7回 UNIXコマンド使いへの道 その5 コマンド行をファイル保存

冬って、行事がとっても多い季節ですよ。クリスマス、忘年会、お正月、新年会、バレンタインデー、卒業式、そして3月下旬には、私のお誕生日もあり、それで長い冬が終わります。なんだか春が待ち遠しくなってきました。

よしだともこ <http://www.tomo.gr.jp/>

My "Happy 物書き Life"

先日、富山大学で「『理系は文章が苦手、文系は文章が得意』というのは本当だろうか～読み手を意識した文書作成術入門～」というタイトルの特別講義¹をしました。この大学では、学生に言語表現力を身に付けてもらうために、全学生に対して「言語表現」と「情報処理」の少なくとも1つを選んで、必修科目としておられるそうで、この「言語表現」教育に携わっている先生の依頼で、話をすることになったというわけでした。

「特別講義は別に単位になるわけでもないし、しかも、言語表現という地味なテーマの講義に、今どきの大学生は集まってくれるのだろうか……」という不安のあった私が、ちょっと客寄的に考えたのが、このタイトルでした。自分としては、なかなか気に入っていたのですが、講義に行く前に、「私、こういうタイトルの講義をしに行くねん」と言った、ほぼ全員の知り合いから、「タイトル、あやしい……」と言われてしまいました。その後、「まあ、そういう傾向はあるのかなあ」とか、「今の大学生は学部に関係なく苦手やで」とか、「理系でも得意な人はいるで」、「文系の人って、だらだら書くのが好きなだけでは？」とか、そういう返答をしてくれました。

事前に、講義概要として、以下のようなものをアナウンスしました。

講義概要

この講義では、主に読み手を意識した文書作成術を議論します。相手の反応を見ながら言葉が選べる会話とは違い、文章は、書き手が一方的に、読者のレベルや期待を予

測して、書く必要があります。

また、一般的に「理系は文章が苦手、文系は文章が得意」と言われることがあります。これは、果たして本当なのでしょう。もし本当なら、なぜ、そのような現象が起こりがちなのかを、講師の経験から分析し、その対処方法を提案します。

議論の後半では、「読者のレベルや期待を予測して、書く内容や順序を決める」を楽しくするための「ゲーム」を実施します。このゲームは、自分が専門としている分野の専門用語の説明文を、3人の読者(同じ研究分野の人間、同じ国に住む小学生、宇宙人)向けに書いてみるという訓練です。

最後に、インターネットによって気軽に情報発信できる今の時代に、特に、この訓練が必要であり、これが文章力をつけるのに効果的な理由を紹介します。

ふたを開けてみると、予想以上の数の学生が集まってくさり、われながらびっくり。残念ながら、この講義の具体的な内容は、ここでは詳しくは書きませんが、そのうち、どこかに書くかも……(笑)

講義では、何度も「読者の顔(層、知識)が明確になると、書きやすくなる。楽しいと思えるよ」と繰り返してしまいました。私がそうなので……。

つまり、私が、雑誌記事や書籍を書いているのは、読んでくれているであろう読者が、私の文章にクスッと笑ってくれたり、「なるほど！」って思ってくれたりする瞬間を勝手に想像して楽しんでいるからなんです。ちなみに、最近はやりの「動物占い」では、私は「サル」でした。サルというのは、「褒められたいために頑張る」傾向があるそうです。そして「器用だったり、だまされやすかったり」もするそうです。げ、げ。

*1 講義について詳しくは、富山大学言語表現部会のページ <http://hyogen.edu.toyama-u.ac.jp/> および、富山大学 向後千春先生の1999年12月16日の日記 <http://kogolab.edu.toyama-u.ac.jp/chiharu/mds/1999/12/16.html> 参照。当日のスナップ写真もあります。

当たってる……。

時々、「子供の頃から、文章を書くのが好きだったの？」と聞かれるのですが、その答えは、「小学校の1年と2年生の時は大好きだった。高校の小論文の授業はすごく苦手だった……」です。自分の考えのない、非常にボーッとした高校生だったので、小論文は全然書けなかったです。友情とか、そういうテーマの文章は恥ずかしくて書けなかったし、政治問題とかは興味なかったし(実は今も……)。

小学校の1年と2年のときに作文が好きだったのは、担任の先生が、私の書く作文をいつもすごく褒めてくださったからです。先生自身が私の作文を読むのが楽しいって、よく言ってくださいました。「きょう、たにしを見つけました。たにしの背中がソフトクリームみたいにとんがってました。」という文章の後半に線が引かれて、花丸が付いてたりして、「私って作文の天才かも……。普通の子供は、たにしを見ても、ソフトクリームとは思わないのかな」と思ったものでした。たぶん、そう思う子供はいくらでもいるけど、あえて作文に書かないだけなんでしょう(-)。

一般的に、人間って、人を喜ばせることが好きな動物ですよ(私だけってことはないと思う……)。同時に、「自分はこんなことができるんだよー」ってさりげなくアピールしたい動物でもある(これだって、私だけってことはないと思う……)。だからこそ、お金なんてもらえなくても、「フリーソフトを書いて公開する」のだろうし、同じように、ぬいぐるみの型紙を公開したり、作った作品を公開したり、作り方を教えたがったり、それにまつわる文章を書きたがったり……。多分テーマはなんでもよくて、「すごいね」、「ありがとう」って言ってもらえることが、満足感を与えると思うんですよ、人間という動物に。

先日、口コミ地域情報(お店、病院、公的施設など)を本にまとめた、主婦のグループの方と話していたら、「この本作りを、みんながボランティアとして取り組んだから、これだけ濃い情報が満載された文章が集まった。これがもしアルバイトだったら、それなりの文章しか出てこなかったと思う。自分がパートにでて時給1000円もらえるとして、もし、原稿書きに対して「400字で250円」みたいな単価をつけたら、15分で400字を書こうとしたと思う。値段がなかったからこそ、丸1日かけて、わざわざ交通費や時間をかけて、店に調査に行ってくれたのだと思う」って言っておられました。なるほどね。

さて、話を戻しまして、上述の講義に集まってくれた、その大学の学生の中には、私の雑誌記事や書籍の読者の方もおられました。それは、明示的に「読んでくれるの?」って聞く前に分かったんです。というのは、文章表現の講義なので、

「難しい概念を説明するときは、例えを使うなどの工夫が必要だ」

みたいな箇所があったので、そこでわざと、「例えば、オープンソースという概念を説明するのに、手作りぬいぐるみを例えとして使ってみたりすると、一般の人にも理解しやすくなるわけです」と言ってみたら、その瞬間に、ニヤッ~としている学生が、数人いたからです。講義が終って聞いてみたら、やっぱり。

しかも、なんと!! わざわざ著書を持ってきて、「サインを」なんて言ってくれた学生や先生もおられて、すごく嬉しかったです。著者にサインを求めるのって、すご~く「著者孝行」だと思いますよ、私は。1冊の本をまとめるのは、はっきり言ってつらい作業で、1冊終ると「もうヤダ。しばらくは書きたくない」って思うことが多い。でも、読者からの感想を受け取ったり、「サインを」なんて言葉が、新しい本を書こうというエネルギーを運んでくれるような気がします……。

結局のところ、なんやかんや言って(実は最近、身近な人に、よく弱音をはいてるんですが……) 私って「物書き」を楽しんでいるんでしょうね。

UNIXコマンド使いへの道 ~ コマンド行をファイル保存 ~

さて、今回のUNIXコマンド使いへの道は、「コマンド行をファイル保存」です。つまり、前回までに、grepやsedのようなコマンドを紹介してきた知識で考えたコマンド行をファイルに保存しておき、より便利に使おうという話です。通常、こういうことを、「シェルスクリプト」って呼ぶんですが、なんか、私が考えた、1行あるいは、ほぼ同じような行が単にずらずら並ぶようなのを、「シェルスクリプト」なんて立派な名前前で呼ぶのは申し訳ない気がして、それで、「コマンド行をファイル保存」という、ダサイ名前をつけてみました。

名前はともかく、つまり、こういう話です。

UNIXコマンドを組み合わせ、あるコマンド行を実行したとします。例えば、

```
$ grep ^From: * | sed 's/^.*From:/' | sort -u
```

は、メールアドレスのリストを得る、便利な使い方です。こういう便利なコマンド行は、毎回、考えながら入力するのではなく、あるファイルに保存しておけば、気軽に使えるから、それをやっつけてしまおう!! という話です。

このコマンド行の意味を説明する前の前提知識として、次のことを知っておくと良いでしょう。UNIX上で一般的に使われているメーラ(MewやMH)で届いたメールを読むと、各自

のホームディレクトリの下のMailというディレクトリの、さらに下にあるinboxというディレクトリに、それぞれのメールが1メール1ファイルとして置かれます。そして、それぞれのファイル名は、数字です。具体的には、こんな感じです。

```
$ cd ; cd Mail/inbox
$ ls -l | head -5
total 896
-rw----- 1 tomo tomogrp 3364 Jun 17 10:19 1
-rw----- 1 tomo tomogrp 4725 Jun 29 11:14 10
-rw----- 1 tomo tomogrp 2531 Aug 29 13:28 100
-rw----- 1 tomo tomogrp 2973 Aug 29 13:28 101
```

そして、例えば、“1”という名前のファイルの内容は、こんなものです(リスト1)。

つまり、それぞれのファイルのFrom行に、そのメールを書いた人のメールアドレスが含まれているということです。

ここまでで紹介した、コマンド行のうち、

```
$ cd ; cd Mail/inbox
```

というのは、

```
$ cd
$ cd Mail/inbox
```

と同じ意味です。つまり、まずホームディレクトリに戻り、次に、Mailディレクトリの下のinboxディレクトリに移動しています。そして、

```
$ grep ^From: * | sed 's/^.*From: //' | sort -u
```

という行が、今回の注目行なのですが、3つの部分から構成されており、それらが、2本のパイプ(|)でつながられています。つまり、grepの結果を、sedの入力とし、sedの結果をsortの入力としているのです(という説明がいまいち分からなくても、安心して下さい。以下に、具体的に説明しますから.....)。

リスト1

```
$ less 1
Received: from tomo.tomo.gr.jp (tomo.tomo.gr.jp [210.172.111.164])
    by biwasv01.biwa.ne.jp (8.9.1a/3.7W) with ESMTP id MAA08307
    for <tomoko@mx.biwa.ne.jp>; Wed, 9 Jun 1999 12:27:00 +0900 (JST)
Received: from dns1.tomo.gr.jp (localhost [127.0.0.1])
    by tomo.tomo.gr.jp (8.8.7/8.8.7) with SMTP id MAA09269;
    Wed, 9 Jun 1999 12:34:34 +0900
From: Tomoko Yoshida <tomo@tomo.gr.jp>
Subject: [hsl:00039] Penguin Neta
```

よしだです。

> ペンギンネタ，続けます．
同じく。

> このあいだ，とあるチャンネルで IRC してたときに，
♪ ペンギンは2.x系カーネルのロゴキャラクター
></usr/src/linux-2.0.36/Documentation/logo.txt
> で，1.xのはカモメだ」という話を聞きました．だから
> 256本，RunRunLinux の表紙がカモメなんだあ，と感動
> してしまいました．

ということは、HSL本のカラーページの

Linuxとペンギン

のところを、一言、「正確には、ペンギンは2.x系カーネルのロゴキャラクターで、1.xではカモメでした。」と書き加えておくと、より親切ですよ。

ちょうど今、HSL本の増刷用の修正点をまとめているのです...

著者注：256本 「Linuxを256倍使うための本(アスキー出版)」、RunRunLinux 「Run Run Linux(アスキー出版)」、HSL本 「ホップ！ステップ！Linux！(翔泳社)」

まず、最初のgrepの部分を実行してみましょう。これは、行の先頭“^”がその意味が、“From:”から始まる文字列を、このディレクトリ以下のすべてのファイル(“*”がその意味)から、探し出しています。その結果、メールの差出人を表示させることができます。

```
$ grep ^From: *
1:From: Tomoko Yoshida <tomo@tomo.gr.jp>
10:From: Hidesuke XXXXX <XXXXX@246.ne.jp>
100:From: Susumu XXXXX <XXXXX@yo.rim.or.jp>
101:From: Tomoko Yoshida <tomo@tomo.gr.jp>
102:From: XXXXX Katsumi <XXXXX@ops.dti.ne.jp>
103:From: Hideo XXXXX <XXXXX@YYYYY.ac.jp>
104:From: Mariko Katagiri <XXXXX@notredame.ac.jp>
105:From: Tomoko Yoshida <tomo@tomo.gr.jp>
:
```

欲しい情報はメールアドレスだけなので、「ファイル番号: From:」の部分は邪魔ですよね。そういうときに役立つのが、sedコマンドです。grepの出力結果を、“|”(パイプ)を使って、sedの入力に渡します。

“sed 's/^.*From:/'”というのは、“^.*From:”の部分をももなしに置換s/検索文字列/置換文字列/)することを指示しています。この“^.*From:”の部分というのは、先頭(“^”がその意味)から、任意の1文字2つと直前文字の0回以上の繰り返しの後に“From:”という文字のある部分(“.*From:”がその意味)と合致します。そのため、邪魔な部分が消えてくれるのです。

```
$ grep ^From: * | sed 's/^.*From:/'
Tomoko Yoshida <tomo@tomo.gr.jp>
Hidesuke XXXXX <XXXXX@246.ne.jp>
Susumu XXXXX <XXXXX@yo.rim.or.jp>
Tomoko Yoshida <tomo@tomo.gr.jp>
XXXXX Katsumi <XXXXX@ops.dti.ne.jp>
Hideo XXXXX <XXXXX@YYYYY.ac.jp>
Mariko Katagiri <XXXXX@notredame.ac.jp>
Tomoko Yoshida <tomo@tomo.gr.jp>
:
```

この出力結果は、メールファイルのファイル順に並んでいるので、「メールアドレスのリストを得たい人にとって、2つの不都合があります。

- 順不同である(見つけにくい)
- 同じ人のメールアドレスが何度も出てくる

そこで、を解決するために、sortコマンドを使い、アルファベット順に並べます。

```
$ grep ^From: * | sed 's/^.*From:/' | sort
Hideo XXXXX <XXXXX@YYYYY.ac.jp>
Hidesuke XXXXX <XXXXX@246.ne.jp>
Mariko Katagiri <XXXXX@notredame.ac.jp>
Susumu XXXXX <XXXXX@yo.rim.or.jp>
Tomoko Yoshida <tomo@tomo.gr.jp>
Tomoko Yoshida <tomo@tomo.gr.jp>
Tomoko Yoshida <tomo@tomo.gr.jp>
XXXXX Katsumi <XXXXX@ops.dti.ne.jp>
```

また、sortコマンドに“-u”オプションを付けると、何度も出てくる行を1行にまとめてくれますから、も解決できます。

```
$ grep ^From: * | sed 's/^.*From:/' | sort -u
Hideo XXXXX <XXXXX@YYYYY.ac.jp>
Hidesuke XXXXX <XXXXX@246.ne.jp>
Mariko Katagiri <XXXXX@notredame.ac.jp>
Susumu XXXXX <XXXXX@yo.rim.or.jp>
Tomoko Yoshida <tomo@tomo.gr.jp>
XXXXX Katsumi <XXXXX@ops.dti.ne.jp>
:
```

こんなふうに、画面にだらだら表示させるのではなく、あるファイル(例えば“hsl_ml”)に保存したいなら、リダイレクション(">"マークがその意味)の後ろにファイル名です。

```
$ grep ^From: * | sed 's/^.*From:/' | sort -u > hsl_ml
$
```

定期的にこのコマンド行を実行したいなら、この行自体を、あるファイル(例えば“mlist”)に保存しておきます。このようにファイルに保存するときは、“> hsl_ml”の部分はない方が汎用的に使えますから、この部分は書かないでおきましょう。結局、mlistの中身はリスト2のようになります。

コマンド行を保存したファイルを、いざ使いたくなったときには、以下のように、“sh”というコマンドの後ろにファイル名を書けばその行が実行できます。

リスト2

```
grep ^From: * | sed 's/^.*From:/' | sort -u
```

```
$ sh mlist > hsl_ml
```

これで、

```
$ grep ^From: * | sed 's/^.*From:/' | sort -u > hsl_ml
```

を実行したのと同じことになります。これが、今回のテーマの「コマンド行をファイル保存」でした。

あと少し、手を加えておけば、「mlist」をまるでUNIXコマンドであるかのように使えます。つまり、以下のような使い方です。

```
$ mlist > hsl_ml
```

そのためには、ファイルの先頭に、「#!/bin/sh」という「おまじない」を書いておきます。これは、「以下の行を、Bシェル(/bin/sh)に渡してね」というおまじないです。それに加えて、chmodコマンドを使って、mlistファイルを実行可能な形式に変えておきます。

```
$ cat mlist
#!/bin/sh
grep ^From: * | sed 's/^.*From:/' | sort -u
$ ls -l mlist
-rw-r--r--  1 tomo  tomogrp  56 Dec 17 06:07 mlist
$ chmod +x mlist ; ls -l mlist
-rwxr-xr-x  1 tomo  tomogrp  56 Dec 17 06:07 mlist
```

こうしておけば、mlistは、まるでUNIXコマンドであるかのように使えるようになります。こんなふうに……。

```
$ mlist > hsl_ml
$ cat hsl_ml
Hideo XXXXX <XXXXX@YYYYY.ac.jp>
Hidesuke XXXXX <XXXXX@246.ne.jp>
Mariko Katagiri <XXXXX@notredame.ac.jp>
Susumu XXXXX <XXXXX@yo.rim.or.jp>
Tomoko Yoshida <tomo@tomo.gr.jp>
XXXXX Katsumi <XXXXX@ops.dti.ne.jp>
:
```

めでたし、めでたし。

私の愛用コマンド行を紹介

さて、私が1年ほど前から愛用しているのが、MH(Message Handler)コマンド群の中の、/usr/local/bin/mh/pickと

/usr/local/bin/mh/refileを使って、届いたメールを、後でまとめて自動的に振り分ける方法です。これは、Mewでメールを読み書きしている人にも、「mh-e」を使っている人にも、共通して便利ははずです。

以前は、自動的に振り分けコマンドprocmailを使い込もうとしていたのですが、これは、読む前に振り分けられてしまうので、どうも私にはあまり使いやすくないなあ……と思っていたところ、Web上に転がっていた古い文書(10年前に書かれたとかいう、そういう古さ)で、この使い方を知りました。

私は、「mh_sort」という名前のファイルの中に、リスト3の行をずらずら書いています。

つまり、まずはすべてinboxに取り込み、シーケンシャルに読み、適当な時点でメールをざっと読み終ってログアウトする前など、「mh_sort」コマンドを実行して、Toフィールド、あるいはFromフィールドを調べて、あらかじめ指定した文字列に合致するものだけ、自動的に分類しているのです。これで、Mailディレクトリの下の各ディレクトリに、メールが分類されていきます。合致しないものが、inboxに残ります。

refileの行は、それぞれ普通のUNIXコマンド行なので、

```
$ refile +tomoko 'pick +inbox -from Tomoko'
```

というふうに、1行だけ取り出して実行することも可能です。私は、まずは1行単位で実行してみて、OKなら、ファイルに記述しています。

なお、「refile」や「pick」の詳しい使い方については、

```
$ man mh
$ man refile
$ man pick
```

で、MHコマンド群のオンラインマニュアルを見て、調べてください。

今回は、これでおしまいです。では、また。

リスト3

```
#!/bin/sh
refile +llug 'pick +inbox -to Linuxer'
refile +freewnn 'pick +inbox -to freewnn'
refile +wnn 'pick +inbox -to wnn-users'
refile +visit 'pick +inbox -to visit2'
refile +hsl 'pick +inbox -to hsl'
refile +lc99 'pick +inbox -to lc99-'
refile +tomoko 'pick +inbox -from Tomoko'
:
```