よしだともこの Linux事始めの書

第4回 UNIXコマンド使いへの道
——その2 「正規表現」徹底入門

秋は運動会、遠足、文化祭と行事の多い季節です。この時期、青くてすっぱい「みかん」がおいしいんですよねー。

よしだともこ　http://www.tomo.gr.jp/

My“Happy 異文化体験 Life”

先日、京大の文学部の中世哲学研究室主催の研究会[*1]におじゃましてきました。なぜに突然、中世哲学の研究会に！？　それは、中世哲学の研究にはなくてはならない「古典ギリシャ語」を、Mule（Emacs 20）上で読み書きできる環境作りに、ほんの少し関係していたからです。その環境が、「古典ギリシア語環境作り[*2]」のリーダーである水落健治さん（明治学院大学で哲学、キリスト教学、ラテン語を教えておられる）によって、この研究会で発表されるということで、声をかけていただけたというわけでした。

研究会の最初の発表は「リチャード・ルフスの認識論」というタイトルのもので、私には当然、理解できませんでした。が、私は、こういうふうに自分だけ1人、ぽつんと取り残される状態が、ぜんぜんイヤじゃない人なので、「うぁ〜、世の中には、こういう研究をしてる人がいるんだ……」と、目の前で繰り広げらる異文化の世界[*3]を楽しませていただきました。

その次が、いよいよ、水落健治さんによる「古代・中世哲学研究におけるコンピュータの利用について——PC-UNIXでのギリシア語を含むマルチリンガル環境の事例を通して」の発表でした。ソニーのVAIOにインストールされたVine Linux上のEmacs 20の画面が、教室のスクリーンに映し出され、まず、日本語、各種ヨーロッパ言語、古典ギリシャ語を混在させて入力する様子がデモされました。

そのようにして入力したマルチリンガル文書は、自動的にLaTeX形式でも出力することができるようになっており、各種の言語を混在させたものが、美しく印刷されている論文が紹介されました。

さらに、現存しているすべての古典ギリシャ語のテキストのデータベース化を目指して活動している、アメリカのTLG（Thesaurus Linguae Graecae）プロジェクトから入手した電子化テキストを、Emacs用に変換させたテキストが、スクリーンのEmacs上に表示されました。なんでも、古典ギリシャ語で書かれた8万行にも及ぶプラトンの全著作を1つのファイルとして扱っていたそうで、用語検索の様子が紹介されると、会場からは、大きな拍手や感動のどよめきが、頻繁にもれるようになっていました。

たぶん、8万行にも及ぶプラトンの著作から、人間が用語を検索して、その用語が含まれる文だけを別ファイルに保存しようとか、用語索引を作ろうとか、そのような作業をすることと比較すると、コンピュータによるテキスト処理は、魔法のように思えたはずです。それに、パソコン上の一般的なエディタやワープロで、8万行のテキストが、あのカーソル移動の速さで扱えるとはとう

---

*1 「中世哲学研究室主催の研究会」については、http://veritas.bun.kyoto-u.ac.jp/saisin.htm参照。

*2 「古典ギリシア語環境」については、http://www.etl.go.jp/~ntakahas/npx/cgreek-emacs20/参照。9月末に発売された『よしだともこのルート訪問記書籍版』には、この紹介記事があり、この書籍の付録CD-ROMには、古典ギリシア語環境作りのためのパッケージも含まれている。

*3 コンピュータの世界でもいまだに、異文化の世界の感動を楽しんでいるような気がします。

ていい思えません。しかも、「日本の普通のコンピュータでは扱えないよ」、「日本語との混在はできないよ」などと、一般的に思われていた古典ギリシャ語のテキストに対する処理が、目の前で、ひょいひょいと扱われていたのですから、会場がどよめくはずです。

この環境作りプロジェクトの原型は、1993年ごろからスタートしており、水落健治さんたち、ギリシャ語やラテン語の電子テキスト化、ファイル内での混在に興味を持つ中世哲学研究家が、電子技術総合研究所のMule開発者の高橋直人さんたちの協力、指導のもとで、本格的にスタートされたのが、3年ほど前でした。古典ギリシャ語のフォントを自分たちで作るところからの長い道のりを思うと、私まで感無量です。

さて、水落さんの発表の後半では「自分がこの環境を手に入れるには、どうすればよいのか」という活発な質疑応答が、繰り広げられました。この環境は、UNIX上のEmacs 20での利用に加えて、Emacs 20をWindowsに移植した、「Meadow」(Multilingual enhancement to gnu Emacs with ADvantages Over Windows)でも使えるパッケージができあがっています。そのため、Windowsユーザーは、まずはこちらの環境作りからスタートして、必要に応じて、Linuxを導入していけばよいだろうという話をしました。別の特殊な言語環境を整える手段についても、議論されました。

この時、目の前で熱く繰り広げられた、文系の研究者たちの質疑応答を聞きながら、「こういう人たちこそが、クライアントとしてのUNIX(Linux)環境を必要としていたのだ！！」という思いを強めました(逆に言うと、「Linuxのインストールには成功しました。次は、何をすればいいのでしょう？」という人々は、Linuxなど、無理して使わなくてもいいのです。すでに使っているOSで、本人が快適だと思える環境が得られているのですから……)。

もともとUNIXには、大量のテキストデータを処理できるコマンドが山ほどあります。日本では、日本語の扱いの問題があったために、文系の研究にはあまり使われてきませんでしたが、欧米の文系の研究には長い歴史があります。しかも、テキストデータ処理のためのUNIXコマンド群の使い方は、ここ数十年、基本的に変っていませんから、一度使いやすい環境を作っておけば(ま、この部分が大変なんですが)、それをずっと使い続けることができます。10年、20年と、長期に及ぶ文系分野の研究で一番困るのは、ワープロなどのバージョンがどんどん変わり、ファイルに互換性がなくなることだそうです。その点は、UNIXをベースにした環境なら安心です。

また、古典ギリシャ語のように、少数の人しか必要でない環境は、商用ソフトウェアの登場を期待することができません。作っても儲かりませんからねぇ……。そこで、自分たちで作るとなると、ソースが公開されている環境の上で作るのが、好都合だというわけです。特にMule(Emacs 20)には、ユーザーが新しい言語環境を加えるしくみが提供されていますから、ますます好都合です。各大学の研究室に、この環境作りのノウハウが蓄積されれば、着実に後輩の研究者に受け継がれていくに違いないでしょう。

研究会の後の懇親会でも、引き続き、このようなことを語り合いました。そこで、若手の研究者たちは、UNIXについて正しい知識があることを知りました。これは、京大では1998年3末までは、センターが提供していたインターネット利用のクライアント環境がUNIXのみだったからで、Muleもすでに使い込んでいるそうです。これは好条件！　彼らが、水落さんたちが培ってきたノウハウが伝わり、この環境作りの試みが、発展し続けることを祈らずにはいられません。

「知らなかったわけではないんですが……」

さて、さて。

この連載では、前回から、「UNIXコマンド使いへの道」というテーマでの話をスタートさせています。特に、テキスト処理関係のコマンドを順番に紹介していく予定で、前回は「grep」を紹介しました。まずは、前回の記述の訂正から。

前回の記事の中で、

「list」というファイルから、「Chap.1」を含む行を取り出すためには、

```
$ grep Chap.1 list
```

と入力すればよい。

と書きました。雑誌の発行日直後、友人から、

表1　基本正規表現の一覧

特殊文字	意味	例	合致する文字列
.	任意の1文字	r.t	rat r2t など
*	直前文字の0回以上の繰り返し	ro*t	rt rot root rooot など
[ ]	[]内の文字のいずれか1文字	Chap[13]	Chap1 Chap3
[-]	同上(範囲指定)	Chap[1-3]	Chap1 Chap2 Chap3
[^]	[]内のいずれでもない1文字	Chap[^13]	Chap2 Chap4 など
^	行頭	^From:	行頭の From:
$	行末	jp$	行末の jp
\	特殊文字の解除	Chap\.1	Chap.1

「.」は任意の1文字とマッチするので、これだと、「ChapA1」も「Chap-1」も「Chap11」も拾っちゃいますよ。ここは、「fgrep」を使うほうがよろしいのでは。

というメールが届きました。う、う、う。言われてみると、まさにその通り。つまり、grepは正規表現(検索文字列中に指定して、特別な意味を持つ文字)を解釈します。そして「.」(ピリオド)は正規表現の1つで、任意の1文字とマッチするという意味なのです(正規表現については表1参照)。

一方、fgrepの方は、書かれたままの文字列を探します。だから、grepだと「list」というファイルに「ChapA1」と「Chap-1」と「Chap11」と「Chap.1」とが含まれる行が1行ずつあったとすると、「Chap.1」以外も拾ってしまうというわけでした。以下が、実行結果です。

```
$ cat list
○○○訪問記の関係者リスト
00:00:00:イラスト:大山正弥:おおやま まさや:Oyama Masaya:ChapA1
00:00:00:編集:渡邊淳子:わたなべ じゅんこ:Watanabe Junko:Chap-1
00:00:00:編著:吉田智子:よしだ ともこ:Yoshida Tomoko:Chap11
02:95:03:KYSA:孝本達哉:こうもと たつや:Kohmoto Tatsuya:chap.1
03:95:04:NDWC:グレッグ ピーターソン:ぐれっぐ ぴーたーそん:Greg Peterson
04:95:05:奈良女:鴨浩靖:かも ひろやす:Kamo Hiroyasu:Chap.1
```

```
$ grep Chap.1 list
00:00:00:イラスト:大山正弥:おおやま まさや:Oyama Masaya:ChapA1
00:00:00:編集:渡邊淳子:わたなべ じゅんこ:Watanabe Junko:Chap-1
00:00:00:編著:吉田智子:よしだ ともこ:Yoshida Tomoko:Chap11
04:95:05:奈良女:鴨浩靖:かも ひろやす:Kamo Hiroyasu:Chap.1
```

たしかに、余分なのまで拾ってしまいましたね。一方、fgrepを使えば、

```
$ fgrep Chap.1 list
04:95:05:奈良女:鴨浩靖:かも ひろやす:Kamo Hiroyasu:Chap.1
```

というふうに、正しく「Chap.1」が含まれる行だけを拾ってくれます。どうしても「grep」を使いたいのなら、正規表現の前に「\」をつけて、「' '」あるいは「" "」でくくります。

```
$ grep 'Chap\.1' list
04:95:05:奈良女:鴨浩靖:かも ひろやす:Kamo Hiroyasu:Chap.1
```

```
$ grep "Chap\.1" list
04:95:05:奈良女:鴨浩靖:かも ひろやす:Kamo Hiroyasu:Chap.1
```

というふうになるわけです。

しかし、こんな基本的なことを間違えるとは、我ながら情けない。読者のみなさん、ごめんなさい。その昔、返却されたテストを親に見せるたびに、「本当は、知ってたんだけど、間違えた」と言って「知ってたなら普通は間違えない。間違えたら知らなかったのと同じ。言い訳してないで勉強しなさい!」としかられた時のことを思い出して、シュンとしてしまいました。

「言い訳してないで勉強しなさい!」それは言えてます。頭が良ければいいのですが、そうでない私には勉強不足は禁物です。そこで今回は、非常に基礎的な勉強からやりなおすことにしました。退屈な人も多いと思いますが、許してください。

grepファミリーのオプションのまとめ

まずは、fgrepを含むgrepファミリーでよく使われるオプションをまとめてみます(表2)。

表2　grepのオプション

-i	大文字と小文字を区別しないで拾ってくれる
-c	検索条件に一致した行数だけを出力
-n	行番号も付加して出力
-l	その検索文字列が含まれているファイル名だけを表示

実際に使ってみましょう。

```
$ fgrep -i Chap.1 list
02:95:03:KYSA:孝本達哉:こうもと たつや:Kohmoto Tatsuya:chap.1
04:95:05:奈良女:鴨浩靖:かも ひろやす:Kamo Hiroyasu:Chap.1
$ fgrep -ic Chap.1 list
2
$ fgrep -in Chap.1 list
6:02:95:03:KYSA:孝本達哉:こうもと たつや:Kohmoto Tatsuya:chap.1
8:04:95:05:奈良女:鴨浩靖:かも ひろやす:Kamo Hiroyasu:Chap.1
$ fgrep -l Chap.1 list
list3
$
```

この実行結果だけを見ていると、「行数だけとか、ファイル名だけとかを表示させて、何が嬉しいの？」という感じですが、UNIXの醍醐味は、この結果を次のコマンドの入力にすること、つまりフィルタとしての使い方です。それについては、次回以降、説明していきます。

また、日本語文字列に対応したgrepを使い、かつ、自分が使っているシェル上の入出力に日本語が使えるようになっていれば、次のように、検索文字列に日本語を使うことができます。

```
$ grep 鴨浩靖 list3
04:95:05:奈良女:鴨浩靖:かも ひろやす:Kamo Hiroyasu:Chap.1
```

このコマンドが、ほんとうに威力を発揮するのは、「ある1冊の本のテキストデータ全体に対して、ある文字列が含まれたファイルを検索する」とか、そんな時です。たとえば……

```
$ cd /cdrom/genko_text
$ grep 鴨浩靖 *
1_3.txt:From: 奈良女子大学 理学部 情報科学科 鴨浩靖(かも ひろやす)
3_2.txt:奈良女子大学の鴨浩靖先生からmlfilt-tiny2というツール [注13]
3_2.txt:tiny-ml-filter2を、奈良女子大学の鴨浩靖さんが改造したもの。
gaiyo.txt:1.3　奈良女子大学、鴨浩靖さん訪問（1995.5月号）
kakokiji_intro.txt:奈良女子大学 理学部 情報科学科 鴨浩靖(かも ひろ
kakokiji_intro.txt:鴨浩靖さんには、1990年ぐらいから技術的なことを教
```

実行例1　man jgrepの結果

```
$ man jgrep
JGREP(1)                                        JGREP(1)

名前　jgrep - 日本語対応 grep
概略　jgrep [ -[cilsvwNJESX.,] ] [ pattern [ files... ]
解説　jgrepは、patternで指定したパターンにマッチする行を
      filesで指定したファイルのなかから検索します。デフォ
      ルトでは、jgrepはマッチした行を表示します。
オプション
      -c  マッチした行数を出力します。
      -i  大文字と小文字の区別を無視します。
      -l  ファイル名のみを表示します。
      -s  エラーメッセージの出力を抑止します。
      -v  pattern にマッチしない行を表示します。
      -w  pattern を単語として検索します。
      -N  検索結果を New JIS (ASCII + JIS X 0208-1983)
          コードで出力します。
      -J  検索結果を JIS (JIS X 0201 + JIS X 0208-1978)
          コードで出力します。
      -E  検索結果を日本語 EUC コードで出力します。
      -S  検索結果をシフト JIS コードで出力します。

(以下、略)
```

「鴨浩靖」という名前が、ある本に6回、登場していることがわかりました。

先に、「日本語文字列に対応したgrepを使い、かつ、自分が使っているシェル上の入出力で日本語が使えるようになっていれば……」と書きました。私の手元にある環境で調べたところ、Plamo Linux 1.4.2 および Vine Linux 1.1CR および、TurboLinux 日本語版 4.2で使えるgrepは日本語文字列に対応しており、シェル上の入出力に日本語が使えるようになってました。

ただし、元ファイルの日本語の文字コードはEUCがデフォルトになっているため、他の文字コードで保存された文書の場合は要注意です。なお、grepの日本語拡張版であるjgrepには、入力、出力の文字コードを指定するオプションもありました(man jgrepを実行した結果は、実行例1参照)。jgrepは、PJEに含まれているコマンドで、Plamo Linux 1.4.2の「お勧め」でインストールした場合には、「/usr/local/jvim2.0/bin/jgrep」にありました。

次回は、「sed」コマンドを紹介する予定です。では、また。