

# 知能情報工学演習 I

柳本 豪一

平成 17 年 5 月 18 日

ここでは UNIX でよく使われるコマンドについて説明をします。コマンドの名前と機能をしっかり理解しましょう。また、これから説明するコマンドはパイプで他のコマンドと組み合わせて使うことでさらに便利な処理が行えます。

この演習を行う前に、`~thy00057/ecsis1/3`のファイルをすべて各自のホームディレクトリにコピーしましょう。

## 1 less

ファイルの中身を見る方法として、`cat` や `more` などのコマンドを使う方法や Emacs などのエディタを使う方法があります。`cat` や `more` はファイルの先頭から最後に向かって表示するだけのコマンドです。Emacs などのエディタはファイル内を自由に移動することができますが、誤ってファイルの中身を変更してしまう可能性があります。ファイル内の閲覧位置を自由に移動できる閲覧のみの機能を持つコマンドとして `less` があります。`less` はファイルの中身を見るときによく使うことになるので、しっかり覚えておきましょう。また、`less` は `man` コマンドで画面を表示するソフトとして設定されていることが多いです。

`less` の使い方は「`less` ファイル名」です。指定されたファイルがテキストファイルのときは、画面にファイルの内容が表示されます。代表的な操作方法を以下に示します。

- `j`  
一行分表示を下にずらす
- `k`  
一行分表示を上にならす
- `q`  
`less` を終了する
- `(行番号)g`  
`(行番号)` から表示する
- `G`  
ファイルの最後に移動する
- `/(文字列)`  
`(文字列)` を検索する

- n

検索の次の候補に移動する

それでは、sample.txt を使って less の動作を確認してみましょう。まず、ファイルの中を自由に移動しながら中身を見てみましょう。つぎに、hidekazu と書かれている部分をファイルの中から探してみましょう。複数あるので、全部見つけるように試みましょう。最後に、man で less のマニュアルを読んでここで説明した以外の機能を自分で調べましょう。

## 2 head と tail

head はファイルの先頭を表示するコマンド、tail はファイルの末尾を表示するコマンドです。プログラムが表示するメッセージを保存しているログファイルでは、新しいメッセージがファイルの最後に追加される場合が多いです。less を用いて、ログファイルを閲覧することも可能ですが、最新のメッセージのみが知りたい場合には、少し大げさになってしまいます。このようなときに、tail を使います。

使い方は、「head ファイル名」、「tail ファイル名」として使います。それぞれのコマンドでは、ファイルの先頭 10 行、末尾 10 行が表示されます。表示する行数を変化させるには、「-n (行数)」というオプションを付け加えて実行します。

それでは、sample.txt を用いて head と tail を実際に使ってみましょう。表示する行数を変化させてみましょう。最後に man で head、tail について調べてみましょう。

## 3 wc

wc はファイルの行数、単語数、文字数を計算するコマンドです。使い方は、「wc ファイル名」と入力します。出力は、行数、単語数、文字数の順番で表示されます。

よく使うオプションとして、「-l」があります。これは、行数だけを表示するコマンドです。他にも、単語数だけを表示する「-w」や文字数だけを表示する「-c」があります。これらのオプションを組み合わせることもできます。

それでは、wc を使って sample.txt を調べてみましょう。最後に man で wc について調べてみましょう。

## 4 sort

sort はファイルの内容をアルファベット順にソートするコマンドです。使い方は、「sort ファイル名」です。sort を使うと、アルファベット順に並べ替えられたファイルの内容が表示されます。

よく使うオプションとして、「-n」、「-r」、「-k (数字)」があります。「-n」はアルファベット順ではなく、数値としてソート、「-r」は逆順にソートされます。「-k (数字)」をオプションとして付けると、(数字) で指定されたフィールド以降をソートのキーとして用います。フィールドとは空白で分割された項目のことです。ソートのキーとは、ソートで大小を比較する項目のことです。

それでは、sort を使って sample.txt の中身をソートしてください。まずは名前 (先頭のフィールド) の順に、つぎは月 (3 番目のフィールド) の順番に並べてみましょう。最後に man で sort の他の使い方について調べてみましょう。

## 5 uniq

uniq は隣接する行が同じものを省いて表示するコマンドです。隣接する行を省く機能しか持っていないので、離れたところに同じ内容の行があったとしても、同じ行とみなされず削除されません。このため、sort で先にファイルの内容をソートしてから、uniq が使われることが多いです。uniq の使い方は、「uniq ファイル名」です。

それでは、実際に uniq を使ってみましょう。sample.txt には同じ行が重複して含まれています。まず、uniq をかける前の行数を調べましょう。つぎに、uniq で隣接する行が同じものを省いたときの行数を調べましょう。そして、sort で先にファイルの内容をソートした後、uniq で隣接した行を省いたときの行数を調べましょう。この 3 つの場合で表示される行数が異なります。実行したコマンドの特徴を考えて違いがなぜ発生したか考えましょう。最後に、man で uniq の他の使い方を調べましょう。

## 6 diff

diff は 2 つのよく似たファイルの内容の違いを表示するコマンドです。例えば、プログラムのファイルと、そのプログラムを一部修正したファイルがあるとき、時間が経った後にはどこが異なっているか分からないことがあります。このようなとき、diff コマンドを使うと異なっている部分だけ表示されるので便利です。使い方は、「diff ファイル 1 ファイル 2」です。

それでは、実際に diff を使ってみましょう。その前に、sample.txt と少し異なったファイルを作りましょう。各自、Emacs を使って、sample.txt の一部を修正し、sample2.txt という名前で保存しましょう。それが済んだら、diff を用いて修正した場所が表示されるか確認してください。最後に、man で diff の使い方を調べましょう。

## 7 grep

grep はファイルから特定の文字列を見つけ出すコマンドです。非常に使う機会の多いコマンドなので、しっかりと使い方を覚えておきましょう。

### 7.1 基本的な使い方

grep の使い方は「grep '(検索文字列)' ファイル名」で、ファイル中に (検索文字列) が含まれる行を表示します。また、パイプを使ってコマンドの結果を検索することができます。このときは、「(コマンド) | grep '(検索文字列)''' と入力します。

それでは、sample.txt 中に含まれる hidekazu という文字列を grep を使って調べてみましょう。つぎに今自分が実行しているプログラムの一覧を見てみましょう。UNIX 上で実行されているプログラムの一覧は、「ps aux」で表示されます。これを参考に、自分が実行しているプログラムだけを grep により取り出してしてみましょう。

検索文字列には正規表現が使えます。正規表現を使いこなすことによって、grep はますます便利に使うことができます。

## 7.2 正規表現

正規表現とは、「いくつかの記号を用いてある（文字列の）パターンを表現する」ものです。今までの説明では、探したい文字列をそのまま入力するしかありません。例えば、sample.txt 中で「最後が 1 で終わる行」や「y で始まる単語を含む行」などを探す場合があります。このような条件を表すために正規表現を用います。それでは、正規表現について詳しく説明していきます。

### 7.2.1 位置の指定

行頭や行末などの位置を表す記号が正規表現には用意されています。行頭は「`^`」で、行末は「`$`」で表します。

例をあげて使い方を説明します。以下では、sample.txt を対象とします。「最後が 1 で終わる行」は、正規表現で「`1$`」となります。「先頭が i で始まる行」は、正規表現で「`^i`」となります。この正規表現を用いて、何も記述されていない行（空行）を表すことができます。空行は「`^$`」で表します。つまり、行頭と行末が一致している行を探すということになります。

### 7.2.2 任意の文字の指定

正規表現では、具体的な文字ではなく、任意の文字を指定することができます。よく使うものとしては、「`.`」、「`[ ]`」、「`*`」、「`?`」があります。

「`.`」は、任意の一文字を表します。例えば、任意の 5 文字は「`.....`」となります。

「`[ ]`」は、特定の 1 文字を表します。例えば、「`[abcde]`」は、a か b か c か d か e の一文字を表します。このような条件は、「`[a-e]`」とも書くことができます。他に、列挙した文字以外を指定するために、「`^`」を用います。これは、「`[ ]`」内部で用いられるときに有効で、行頭の「`^`」とは異なります。つまり、「`^[abcde]`」は、a、b、c、d、e 以外の一文字を表します。

「`*`」は、直前の文字の 0 回以上の繰り返しを表します。例えば、「`i*`」は、i を 0 回以上繰り返した文字列を表します。つまり、i が一度もない場合も含まれます。では、i が 1 つ以上続いている文字列はどのように表すのか考えてみましょう。正解は「`ii*`」となります。

「`?`」は、直前の文字の 0 回または 1 回の繰り返しを表します。例えば、「`i?`」は、「`i`」（何もない）か「`i`」を表します。これは、`*`の繰り返し回数が制限されているものと見なせます。

他にもよく使うものとして「`.*`」があります。これは、任意の文字列の繰り返しを表します。例えば、「h で始まって u で終わる文字列」はどのように表すのでしょうか？今まで説明してきた正規表現を用いると、「`h.*u`」となります。

### 7.2.3 正規表現のまとめ

それでは、今まで説明してきた正規表現を使っているいろいろな文字列を表現してみましょう。検索対象のファイルは、sample.txt です。

h が y で始まる行を正規表現で表してみましょう。今までの知識を用いると、行の始まりは「`^`」で表します。「h が y」は、「`[hy]`」で表すことができます。よって、正規表現では、「`^[hy]`」となります。

つぎに数字の 1 で終わるものを探しましょう。今の場合は 11 や 21 で終わるものは検索対象外とします。sample.txt では、空白で数字が区切られているので、数字の 1 は、空白と 1 を組み合わせた「 `1`」と表されます。したがって、正規表現では、「 `1$`」となります。

正規表現は非常に便利です。自分でいろんな表現ができるように勉強してください。

## 8 find

grep はファイルの中身を検索するコマンドでした。find はディレクトリ内に存在するファイルを探すコマンドです。

find の基本的な使い方は、「find (検索対象ディレクトリ) -name (探したいファイル名) -print」となります。find では、指定した検索対象ディレクトリより下のディレクトリも自動的に調べてくれます。このため、ルートディレクトリを対象として検索を行うと全てのディレクトリを検索することになります。

それでは、実際に find を使ってみましょう。/usr ディレクトリ以下に httpd というファイルがあるかどうかを調べてみましょう。コマンドは、find /usr -name httpd -print となります。また、ファイル名にはシェルのメタキャラクタ (詳しくは来週説明します。正規表現に似たものと思っておいてください。) が使えます。例えば、font という文字が含まれるファイルを探したい場合には、find /usr -name 'font\*' -print となります。ここで「'」を忘れないように気を付けてください。最後に、man で find の使い方を調べましょう。

## 9 tar

tar は、ディレクトリの構成を保持したままファイル群を保存する方法です。昔は、データをバックアップするためにテープを用いていたので、そのテープにデータを保存するコマンドが tar です。今では、ディレクトリ構造を持ったまままとめたファイルを作成したり、元に戻したりするために、tar が用いられます。UNIX で使うソフトの大部分はソースファイルが公開されています。このようなソースファイルは、tar でまとめられて公開されています。

ファイルをまとめる方法は、「tar cvf ファイル名 ディレクトリ名」となります。これを実行すると、ディレクトリをまとめたファイルがファイル名として保存されます。

ファイルを元に戻す方法は、「tar xvf ファイル名」となります。カレントディレクトリにファイルの中身が展開されます。

## 10 gzip

gzip は、ファイルを圧縮・展開するコマンドです。UNIX では、一般的に使われるツールなので覚えておきましょう。

圧縮するときには、「gzip ファイル名」として実行します。こうすると、ファイル名.gz というファイルが作成されます。これは、ファイルを圧縮したものです。

展開するときには、「gzip -d ファイル名」として実行します。こうすると、展開されたファイルが出力されます。

それでは、tar と gzip を実際に使ってみましょう。class.tar.gz というファイルを展開してみましょう。無事展開できれば、展開されたディレクトリを tar と gzip を用いて圧縮してみましょう。まだまだいろいろな使い方があるので、各自 man で tar と gzip の使い方を調べてみましょう。

## 11 ssh

ssh は、他の UNIX マシンにログインするためのコマンドです。このような操作をリモートログインと呼びます。UNIX では、他の UNIX マシンにリモートログインして作業をすることができます。ssh はセキュリティ対応をしたコマンドであり、できるだけ ssh を利用するようにしましょう。環境によっては、ssh が利用できない場合がありますが、そのときには telnet などを使うことになります。演習室の環境では telnet は利用できず、ssh でしかリモートログインすることができません。

使い方は、「ssh (ユーザ名)@(マシン名)」です。(ユーザ名) を省略した場合には、現在ログインしているログイン名が利用されます。接続ができると、パスワードを聞いてくるので、自分のパスワードを入力すればリモートログインが完了します。

ログアウトするときは、コントロールを押しながら *d* を押すか、logout と入力します。必ず、上記の方法でログアウトしてから電源を切るなどの操作をするようにしてください。

それでは、隣の人が使っている UNIX マシンにリモートログインしてみましょう。自分のパソコンに入っているユーザは who により表示されます。自分以外のユーザがリモートログインしている時には、そのユーザの名前が表示されるはずですが、