データベース入門 資料 2
テキストファイルの閲覧と文字列検索 (less と grep)

平成 23 年 4 月 25 日

目 次

1 less を使ったファイル閲覧と文字列検索 **1**

1.1 ページャーと less 1
1.2 教材テキスト 1
1.3 less の起動法 1
1.4 less の主要コマンド 2
1.5 練習 2

2 grep を用いた特定行（レコード）の抽出 **2**

2.1 grep の基本的な使い方 3
2.2 grep の出力を less で読む 3
2.3 grep の一般書式 3
2.4 練習問題 4

3 少し高度な検索パターンの指定法—正規表現 **4**

1 less を使ったファイル閲覧と文字列検索

1.1 ページャーと less

テキストファイルの中身を閲覧する道具として，cat コマンドやエディタがあるのは知っているはずですが，その他にページャー (pager) という種類のコマンドがあります。ページャーを使えば，比較的大きなテキストファイルの中身を効率よく閲覧することができます。代表的なページャーとして more や less 等がありますが，ここでは less を使ってファイルを閲覧します。さらにテキストファイル内から必要な情報を取得するために検索操作を行ってみます。

なお，man コマンドを用いてコマンドのマニュアルを閲覧する際，マニュアルの表示には less が使われることが多いです。また，この授業の後半で使うデータベースソフトウェアにおいても，検索結果の表示に less を使います。従って，man コマンドやデータベースソフトウェアを使うためにも less の 基本的な使い方を習得する必要があります。

1.2 教材テキスト

ここで扱うテキストファイルは，新聞記事から頻出単語約 60,000 語を抽出したテキストファイルです。このファイルは次の場所にあります。

ディレクトリの絶対パス名 /pub/db/data

このディレクトリ内のファイル 75.60k.vocab.romaji が当該ファイルです。このファイルには，各行につき単語の

- 表記
- カタカナ表記の読み
- 活用語の見出し語
- 品詞タグ (品詞を数値でコード化したもの)
- ローマ字表記の読み

が記述されていて，各項目は + で区切られています。

1.3 less の起動法

less の基本的な起動法は

`less` *`filename`*

です。もちろん，*filename* には，内容を見たいファイルの名前を指定します。

1.4 less の主要コマンド

下記の各コマンドは，less でテキスト文書を閲覧している最中に使える，less が持っているコマンドです。(シェルの) コマンドラインから使うものではありません。

機能	コマンド	コメント
終了	**q** (または **:q**)	
ヘルプ	**h**	
ページ移動	`<SPACE>` (または CTRL-f または **f**)	次ページに移動する
	b (または CTRL-b)	前ページに移動する
	G	最後の行に移動する
	*n***G**	第 *n* 行に移動する
テキストを検索する	/*pattern* `<Enter>`	*pattern* が最初に現れる位置に移動する (下方検索)
	?*pattern* `<Enter>`	*pattern* が最初に現れる位置に移動する (上方検索)
	n	一番最近行った検索を繰り返す
	N	一番最近行った検索を逆方向に繰り返す

1.5 練習

1. まず，ファイル 75.60k.vocab.romaji の内容を表示するために cat コマンドを実行してみましょう。実行後、適当なところで CTRL-c を押して，cat コマンドを終了させてください。

2. less コマンドでファイルの中身を見てみましょう。less では，`<SPACE>` を押して，次のページを見ることができます。その他の操作は 1.4 節の less の主要コマンドを参照してください。

3. カイバという単語を漢字ではどう記すのでしょう。`kaiba` を検索語として検索してみましょう。less で検索するには / に続いて検索語 (pattern) を記入し `<Enter>` を押します。検索語の記入途中でタイプミスをしたら，CTRL-c で検索語入力を中止できます。

4. 新聞記事に現れた「教育」を含む単語にはどのようなものがあるでしょう。検索を繰り返して調べてみましょう (n コマンドを使う)。検索語は `kyoiku` です。

5. 時間があれば，`man less` で less のコマンドを調べたり，less の操作をいろいろと試してみましょう。

2 grep を用いた特定行（レコード）の抽出

この章では，grep コマンドを用いて，テキストファイルから特定の文字列を含む行のみを取り出して，閲覧する方法を学ぶ。

2.1 grep の基本的な使い方

grep の基本的な実行の形式は次のとおりである。

```
grep 文字列 ファイル名
```

この形式では，grep はファイルの内容から，特定の文字列を含む行のみを出力する。

実行例 以下の実行例において，ファイル名の指定には，tcsh シェルの入力補完機能[1]を使おう。
まず，asshukukuki （圧縮空気）を含む行だけをファイルから取り出してみる。

```
grep asshukukuki /pub/db/data/75.60k.vocab.romaji
```

こんどは asshuku （圧縮）を grep してみる。

```
grep asshuku /pub/db/data/75.60k.vocab.romaji
```

「圧縮 (asshuku)」は欲しいけど「合宿 (gasshuku)」はいらないならば，教材テキストでの項目（列）区切りが +（プラス記号）であることを利用して，

```
grep +asshuku /pub/db/data/75.60k.vocab.romaji
```

を実行すればよい。

2.2 grep の出力を less で読む

「あめ」を調べたいと思って

```
grep ame /pub/db/data/75.60k.vocab.romaji
```

とすると，407 単語（407 行）も該当してしまい，結果を画面に表示しきれない。grep の出力を，パイプ (|) を使って less に入力すると，結果を less で 1 ページずつ見ることができる。

```
grep ame /pub/db/data/75.60k.vocab.romaji | less
```

less の終了方法などについては，第 1.4 節の less の主要コマンドなどを参照のこと。

2.3 grep の一般書式

grep [options] *pattern* [*file...*]

grep は *file* で名前を指定された入力ファイル (*file* が指定されてないか， *file* の部分に - が指定された場合は標準入力) を読み込み，与えられた *pattern* にマッチする部分を含む行を探す。
以下に，しばしば使うオプションを挙げるので，試してみよう。さらに詳しく知りたいときは `man grep` を実行しよう。

[1] `http://echoes.hak.hokkyodai.ac.jp/db/550/?id=20082`

よく使う grep のオプション

`-n`：各出力行の前に，入力ファイルにおける行番号を表示する。
`-r`：ディレクトリ下のすべてのファイルを再帰的に読み取る。
`-v`：マッチした行を表示しない。（マッチしない行を表示）
`--help`：簡単なヘルプメッセージを出力する。
`--version`：grep コマンドのバージョンを出力する。

`--help` と `--version` は多くのコマンドで共通に使えるオプションであり，このオプションを使う際には引数の *pattern* は不要。

2.4 練習問題

1. 教材テキスト (75.60k.vocab.romaji) から yuki を含む行のみを表示しなさい。
2. grep のオプションを使って，教材テキストに yuki を含む行が何行あるかを表示しなさい。必要なオプションは man コマンドで調べること。
3. 教材テキストのうち，ame を含む行のみを，リダイレクト (>) を使ってファイルに格納しなさい。ただし，格納先のファイルは，ホームディレクトリに存在するディレクトリ dbintro の下の ame とする。dbintro が存在しなければ，まず作成すること。
4. ファイル ame の中から 44 を含まない行のみを抽出し，パイプと less で閲覧しなさい。

3 少し高度な検索パターンの指定法—正規表現

grep の *pattern* 引数や less における */pattern* 等では，正規表現 (regular expression) と呼ばれる検索パターンを指定できる。その代表的なものを以下に示す。これ以外の正規表現については，grep のマニュアル等を参照のこと。

`^`	行の先頭
`$`	行の終わり
`.`	任意の一文字
`[...]`	... のうちの任意の一文字。`a-z` や `0-9` のような範囲指定も有効
`[^...]`	... にない任意の一文字。`a-z` や `0-9` のような範囲指定も有効
r`*`	ゼロ回以上の *r* の繰り返し。2 文字以上からなるパターン *str* の繰り返しを指定したければ (*str*)`*`
c	文字 *c* の特殊な意味をなくす

次の例を実行して，簡単な正規表現の使い方を確認してください。

```
grep kuki /pub/db/data/75.60k.vocab.romaji
grep n..kuki /pub/db/data/75.60k.vocab.romaji
grep kuki$ /pub/db/data/75.60k.vocab.romaji
grep 's.*kuki$' /pub/db/data/75.60k.vocab.romaji
```

`.*` は長さが 0 以上の任意の文字列にマッチするので，最後の例は，s を含み行末が kuki である行 (パターン `s.*kuki$` を含む行) のみを出力する。

注意 1　正規表現に使われる特殊な文字（メタキャラクタ）と，シェルのメタキャラクタ (ファイル名に展開される) では，一般に意味が異なる。

注意 2　シェルのメタキャラクタでもある * 等を，正規表現として grep の引数に与えるには，シェルによる * の展開を抑止（エスケープ）する必要がある。