

# テキストファイルと vi エディタ

## 目次

1	テキストファイルとエディタ	1
2	vi エディタを使ってみよう	1
2.1	vi の基本操作	1
2.1.1	vi の起動法	1
2.1.2	vi の編集モード	1
2.2	例題	2
2.2.1	例題 1 —ファイルの新規作成	2
2.2.2	例題 2 —既存ファイルの編集	3
2.3	練習	4
2.4	vi の主要コマンド	5

# 1 テキストファイルとエディタ

cat コマンドや more コマンド等で内容を表示できる、文字の書かれたファイルをテキストファイル (text file) といいます。一方で、cat コマンド等では中身を読めない通常のファイルもあります。これをバイナリファイル (binary file) と呼びます<sup>1</sup>。例えば、画像や音声の入ったファイルの多くや、Windows 上のワープロ等で“普通に”保存したファイルの多くはバイナリファイルです。

UNIX では文書等のデータを基本的にテキストファイルとして保存します。その理由にテキストファイルの持つ汎用性が挙げられます。テキストファイルは、多くのコマンドやワープロ等のアプリケーションで共通に使えるファイル形式ですので、テキストファイルに対しては各種コマンドやアプリケーションによる様々な処理を施せます。

また UNIX 系の OS では、OS 自体の諸設定や各種アプリケーションの動作環境を、テキストファイルに記述することが多いです。コンピュータプログラミングをするにもテキストファイルを作成しなければなりません。そのため、テキストファイルを作成したり編集する道具を使えることは大切です。

さて、テキストファイルを作成したり編集するためのプログラムをテキストエディタ (text editor) といいます。単にエディタと呼ぶことも多いです。UNIX にはさまざまなエディタがありますが、この資料の後の章では UNIX の標準的なテキストエディタである vi の基本的な使い方をご紹介します。

## 2 vi エディタを使ってみよう

### 2.1 vi の基本操作

#### 2.1.1 vi の起動法

vi を起動するには、コマンド行で

```
vi file
```

を実行します<sup>2</sup>。ここで *file* には、新規に作成するファイルの名前や、編集したい既存のファイル名を与えます<sup>3</sup>。

#### 2.1.2 vi の編集モード

vi はモードという概念を持つエディタです。vi では、現在どのモードになっているかにより、使用可能な操作が異なります。

コマンド入力モード vi 起動直後には、このモードになっています。カーソル移動や、文字の消去、ファイルへの保存等の操作は、コマンド入力モードにおいて、vi のコマンドを入力することにより行います。vi のコマンドの一部を第 2.4 節に挙げますので、参考にしてください。

<sup>1</sup>より正確にいうと、文字コードと若干の制御コードのみを含むファイルがテキストファイルであり、それ以外のファイルがバイナリファイルです。cat コマンドを使うと、ファイル内の文字コードが文字に変換されて表示されますが、od というコマンドを使うとファイル内の文字コードそのものを見ることができます。

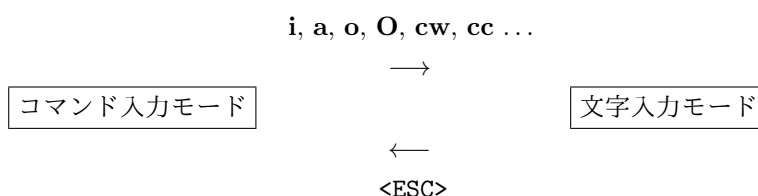
<sup>2</sup>最近のシステムでは vi というコマンドを実行すると、実際には vi の機能拡張版である vim が起動するものが多いです。また、vi コマンドで vi が起動しない場合、代わりに vim と打ってみてください。

<sup>3</sup>*file* を指定せずに vi を起動してから、既存のファイルを読み込んだりもできますが、ここでは必ずファイル名を指定して起動することにします。また、ファイル名を複数指定して編集する機能もありません。

コマンド入力モードでは文字の入力はできません。**i** や **a** 等の **vi** のコマンドを入力し、次に紹介する文字入力モードに移行してから、文字入力を行います。

**文字入力モード** このモードは文字を入力するためのモードであり、基本的にそれ以外の操作はできません。カーソル移動等のためには、**<ESC>** を押して、コマンド入力モードに戻る必要があります<sup>4</sup>。

ただし、**<ENTER>** や **<ESC>** を押す前であれば、入力したばかりの文字は **<BS>** (バックスペースキー) または **CTRL-h** で削除できます<sup>5</sup>。**<BS>** を続けて押すと、さらにその前に入力した文字を削除できます。なお、**<BS>** を押しても、カーソルが左に動くだけで元の文字は表示されたままになっているかもしれませんが、これらは無視してください。**<ESC>** を押せばちゃんと消えます。



**vi** では、これらのモードを行ったり来たりしながら、文書を編集します<sup>6</sup>。現在、どちらのモードなのかが分からなくなったら、**<ESC>** を 1 回または 2 回押して、コマンド入力モードに移行してください。

## 2.2 例題

### 2.2.1 例題 1 —ファイルの新規作成

**vi** を使ってファイルの中身が

```
cal date who
```

である **commands** という名前のファイルを作成してみます。

1. **vi** の起動法に従って、コマンド行に **vi commands** と入力して、**vi** を起動してください。
2. **i** を押してください。

この操作によって「コマンド入力モード」から「文字入力モード」に移行しました。これ以降、カーソル位置に文字を入力できます。

3. ファイルに書き込む内容 (**cal<ENTER>date<ENTER>who**) を入力してください。**<ENTER>** (エンターキー) は改行するためのキーです。タイプミスをしてしまっても、ここでは修正しないでおきましょう。
4. **<ESC>** (エスケープキー) を押してください。

この操作によって、「文字入力モード」から「コマンド入力モード」に移行しました。

<sup>4</sup>文字入力モードでもカーソルキー (矢印キー) でカーソル移動が可能な場合があります。

<sup>5</sup>これらと違うキーを使うシステムもあるかもしれません。

<sup>6</sup>より多くのモードを考えるのがよい場合もありますが、ここでは簡略化してモードは二つとします。

5. **:x** と打って **<ENTER>** を押してください。この vi のコマンドによって、3. で入力した内容がファイル `commands` に保存され、vi が終了してプロンプトが現れます。うまくいかない場合、再度 4. と 5. を行ってください。

以上を終えたら、`commands` という名のファイルが出来上がったことを、`ls` コマンドを使って確認してください。また、`cat` コマンドでファイルの内容を確認してください。

### 2.2.2 例題 2 — 既存ファイルの編集

ファイル `commands` の内容を

```
cal    displays a calender
date   set date and time
```

に変更します。

1. vi の起動法に従って、コマンド行に `vi commands` と入力して vi を起動してください。既存のファイル `commands` の内容が表示されます。  
もし、画面に `commands` の内容が表示されなければ、**:x <ENTER>** を打って一旦 vi を終了し、ファイル名を確認してから、再度 vi を起動してください。
2. 文字を追加すべき位置 (`cal` の `l` の位置) までカーソルを移動します。カーソル移動の方法は、第 2.4 節を参照してください。I (エル) の利用がお勧めです<sup>7</sup>。
3. カーソルの右側に文字を追加するために、**a** を打ちます。  
これにより、文字入力モードに移行しました。コマンド `i` と `a` の違いを第 2.4 節で確認してください。
4. 追加すべき文字「 `displays a calender`」(4 個の空白に続いて `display...`) をタイプします。入力ミスは **<BS>** で修正できます。
5. **<ESC>** を押してコマンド入力モードに戻り、`date` の行の行末までカーソルを移動します。
6. **a** を押して文字入力モードに入り、追加すべき文字を入力します。入力を終わったら **<ESC>** でコマンド入力モードに戻ります。
7. カーソルを `who` の行の行頭まで移動させてください。  
これから `who` を削除しますが、**<BS>** では文字入力モードで打ち込んだばかりの文字しか消せませんので、ここでは文字削除のコマンドを使います。まず **x** を押してカーソルの乗っている文字 `w` を消してみましよう。  
一行すべてを削除するコマンドもあります。**dd** と打って、カーソルの乗っている 3 行目を消してください。
8. 編集を終えたら **:x <ENTER>** を打ちます。変更した内容が元のファイルに保存され、vi が終了します。

---

<sup>7</sup>vi では、右手ホームポジション付近のキーだけでカーソル移動ができます。慣れると快適です。

## 2.3 練習

1. ファイル `commands` の内容を

```
cal      displays a calender
date     display or set date and time
```

に変更してください。

この練習では、ファイルの内容を、上記の通りにきれいに整えましょう。コマンド説明の出だしが揃っていない場合は、揃えてください。不要な空白行があれば、行削除の `vi` コマンドで削除してください。

注意: `vi` は行単位でテキストを編集するエディタなので、改行文字を消すことで複数の行を一行にまとめたりはできません。その代わりにコマンド `J` を使います。

2. `vi` におけるカーソル移動 (`j` や `k` など) やテキスト検索 (`/` や `n` など) の操作は、`man` でマニュアルを閲覧しているときのページ操作と共通です。特に検索の方法は知っておくと便利です。

まず `ls` と `ls -t` を実行して `-t` の効果を予想しましょう。次に `man ls` を実行してオプション `-t` の意味を調べてください。その際には `/` を使って `-t` を検索してください。`ls` のマニュアルには `-t` の記述が複数ありますので `n` を使って検索を繰り返してみてください。

3. 実習用のコンピュータで `vi` を起動すると、実際には `vi` の拡張版である `vim` が起動します。`vim` には操作を学ぶためのチュートリアルがあります。`vimtutor` というコマンドで起動できますので、時間があれば試してください。

## 2.4 vi の主要コマンド

機能	コマンド	コメント
テキストを入力する	<b>i</b> <i>text</i> <ESC>	カーソルの位置に <i>text</i> を挿入する (insert)
	<b>a</b> <i>text</i> <ESC>	カーソルの右に <i>text</i> を追加する (append)
	<b>o</b> <i>text</i> <ESC>	今いる (カーソルがある) 行の下に <i>text</i> を入れる
	<b>O</b> <i>text</i> <ESC>	今いる行の上に <i>text</i> を入れる
ファイルを保存する	<b>:w</b> <ENTER>	編集中のファイルを元の名前のまま保存する
	<b>:w</b> <i>file</i> <ENTER>	編集中のファイルを <i>file</i> として保存する
vi を出る	<b>:x</b> <ENTER> ( <b>ZZ</b> )	ファイルを保存し vi を出る
	<b>:q!</b> <ENTER>	ファイルを保存せず vi を出る
カーソルを動かす	<b>h</b> (←)	左隣に移動する
	<b>l</b> エル (<SPACE> または →)	右隣に移動する
	<b>k</b> (↑)	上に移動する
	<b>j</b> (<ENTER> または ↓)	下に移動する
	<b>0</b> ゼロ	行頭に移動する
	<b>\$</b>	行末に移動する
	<b>CTRL-f</b> <b>CTRL-b</b>	次ページ / 前ページに移動する
	<b>G</b>	最後の行に移動する
	<b>n G</b>	第 <i>n</i> 行に移動する
テキストを削除する	<b>x</b>	今いる文字 (カーソルが乗っている文字) を削除する
	<b>X</b>	カーソルの左隣の文字を削除する
	<b>dw</b>	今いる単語を削除する
	<b>dd</b>	今いる行を削除する
行を連結する	<b>J</b>	今いる行に次の行を連結する
テキストを検索する	<b>/</b> <i>string</i> <ENTER>	<i>string</i> が最初に現れる位置にカーソルを移動する
	<b>n</b>	一番最近行った検索を繰り返す
テキストを置換する	<b>r</b> <i>character</i>	今いる文字を <i>character</i> で置換する
	<b>cw</b> <i>text</i> <ESC>	今いる単語を <i>text</i> で置換する
	<b>cc</b> <i>text</i> <ESC>	今いる行を <i>text</i> で置換する
変更を繰り返す	<b>.</b>	直前のコマンドによる変更を繰り返す
変更を取り消す	<b>u</b>	直前のコマンドによる変更を取り消す (undo)
テキストをコピーする	<b>yy</b>	今いる行を名前なしバッファにコピー (ヤंक) する
	<b>p</b>	名前なしバッファ中のテキストを挿入 (プット) する

括弧 () は当該操作を他のコマンドで行えることを表します。そのコマンドを括弧内に併記しています。  
 斜体字 (*italic*) の箇所は具体的なものに置き換えて記述します。例えば、*text* には入力テキストを、*file* には  
 ファイル名を書きます。