

Emacs で OCaml を使う

設定ファイルの編集

まず, `caml-mode` を使えるようにしなければならない。

情報メディア教育センターの計算機では以下のコマンドを実行すればいい¹。

```
$ cat /mdhome/share/Linux/lib/ocaml/emacs/dot.emacs >> .emacs.el
```

>> に気を付けて下さい。これで `.emacs.el` に `dot.emacs` の中身が追加される。もしも > だけを使えば, `.emacs.el` の元の中身が消えてしまう。

同様に多元数理科学研究科の計算機では以下のコマンドを実行する。

```
$ cat /usr/local/lib/ocaml-3.08.3/emacs/dot.emacs >> .emacs.personal
```

Emacs の起動

モードの全ての機能を利用するには, GNU Emacs 21 を使うのが最もよい。情報メディア教育センターでは単に `emacs` とシェルで入力すればよい。多元数理科学研究科では `emacs-21.4` と入力する。

OCaml を Emacs の中で実行

Emacs の中で `ocaml` を実行するために, 以下でキー列をを入力する。

```
<M-x>run-caml<ret><ret>
```

これで新しいバッファの中で以下の内容が表われる。

```
Objective Caml version 3.08.4
```

```
#
```

の後にプログラムを入れると, そのまま実行される。

```
# let x = 2+2;;  
val x : int = 4
```

このモードで使える主なキー列は以下のとおりである。

<M-p>	以前の入力文を編集する
<C-c><C-c>	実行を途中で中断させる
<C-c><C-d>	ocaml 自体を終わらせる

`ocaml` を直接にシェルで起動することもできるが, そうすると編集機能が使えない。

¹emacs が正しく動かない人は, 設定を編集する前に以下のコマンドを実行して下さい。

```
$ cp /mdhome/pub/center/.emacs.el ~
```

プログラムを編集する

まず、名前が.mlで終わるファイルを作る。

```
<C-x><C-f>test.ml<ret>
```

そのバッファの中でプログラムを書くと、<tab>を押すだけでインデントが自動的に行われる。(構文によって、行を書いてから<tab>を押さないといけない。) また、emacs 21ではキーワードに色が付く。

編集中のプログラムを一段落ずつocamlに実行させることもできる。まず、ocamlを前面に持ってくる。

```
<C-c><C-s>
```

そして、例えば以下の行を書いたら(_はカーソルの位置を表す)

```
let x = 3 * 5;;_
```

今度は次のキー列を入力する(先頭の<C-a>はプログラムの中に戻るため)

```
<C-a><C-c><C-e>
```

そうする実行の結果がocamlのバッファに表れる。(実行したコードがそちらで表示されないの、先頭の#だけが見える)

```
# val x : int = 15
```

もしもプログラムにエラーがあれば、カーソルがその位置に移る。

ファイルからプログラムを読み込む

Emacsのバッファからの評価は中々便利であるが、ファイルを丸ごと読み込むこともできる。これはトップレベルの機能であり、Emacsを使わなくてもできる。そのとき、Camlは、ファイルの内容があたかも入力ループで入力されたように動作する。

ファイルtest.mlの中身は次の通りだとする。

```
let double x = x * 2;;      (* double は引数の2倍を計算する *)
let y = 10;;               (* y を適当な値に *)
y + double y;;            (* これで3倍だ! *)
```

test.mlを読み込む。

```
# #use "test.ml";;
val double : int -> int = <fun>
val y : int = 10
- : int = 30
```

このようにファイルからプログラムを読み込む場合は、入力ループで#use "ファイル名";;のように入力すればよい。答は、読み込んだ入力によるものである。#useは結果を出さない。