Preface

This is a book about hacking in ocaml. It's assumed that you already understand the underlying theory. Happy hacking Most parts are filled with code blocks, I will add some comments in the future. Still a book in progress. Don't distribute it.

☺

Acknowledgements

write later

Contents

Todo list

Chapter 1

Tool Chain

Chapter 2

Lexing

Chapter 3

Parsing

Chapter 4

Camlp4

Camlp4 stands for Preprocess-Pretty-Printer for `OCaml`, it's extremely powerful and hard to grasp as well. It is a source-to-source level translation tool.

Chapter 5

Libraries

5.17 Modules

Chapter 6

Runtime

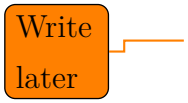
Chapter 7

GC

Should be re-written later

Chapter 8

Object-oriented


Write
later

Chapter 9

Language Features

9.9 The module Language

write later

Chapter 10

subtle bugs

Chapter 11

Interoperating With C

Write later

Chapter 12

Pearls

Chapter 13

Compiler

If you have *Godi* installed, then you have already compiler libs installed.

```
ls `ocamlfind ocamlc -where`/../compiler-lib
```

You can play with compiler lib in the toplevel

```
#directory ``+../compiler-lib'' ;; (** for cmi file*)
#load ``toplevellib.cma'';;
```

Listing 58: Compiler lib in toplevel

13.1 module Printtyp

This module mainly export some printting functions for ocaml ast.

```
val longident: formatter -> Longident.t -> unit
val ident: formatter -> Ident.t -> unit
val type_expr: formatter -> type_expr -> unit
val modtype: formatter -> module_type -> unit
val signature: formatter -> signature -> unit
val tree_of_modtype_declaration: Ident.t -> modtype_declaration -> out_sig_item
val modtype_declaration: Ident.t -> formatter -> modtype_declaration -> unit
val class_type: formatter -> class_type -> unit
```

Listing 59: Printer in compiler lib

You can use this library to process *cmi* files, The structure of *cmi* file is organized as follows

```
let ic = open_in_bin filename in
let magic_len = String.length (Config.cmi_magic_number) in
let buffer = String.create magic_len in
really_input ic buffer 0 magic_len ;
let (name, (sign:Types.signature)) = input_value ic in
let (crcs : (string * Digest.t) list) = input_value ic in
let (flags : flags list) = input_value ic in
close_in ic ;
```

Listing 60: Structure of cmi file

Chapter 14

XX

Chapter 14

Topics