

Initialiser OPAM

Yoan Gérard, Amélie Ledein & Louis Lemonnier

1 Programmer en OCAML

Les éditeurs de texte sont tous très extensibles. Si bien que la plupart des langages de programmation disposent d'un mode pour aider à développer. Ces modes proposent en général de la coloration syntaxique, de l'indentation automatique et de la complétion.

OCAML est un langage fortement typé, ce qui permet d'éviter un grand nombre d'erreurs de programmation. En temps normal, il faut appeler le compilateur à chaque fois que l'on veut vérifier le code écrit. Ce qui devient vite fastidieux.

Le programme MERLIN permet aux éditeurs de vérifier le types des expressions entrées dans le fichier, sans avoir à le compiler. Il permet en outre de donner le type d'une expression à la volée.

Il est possible de l'installer avec OPAM, le gestionnaire de paquets pour OCAML. Il est normalement installé, mais il faut le configurer. Lisez la suite de ce document pour configurer OPAM

1. lancez `$ opam init`
2. OPAM devrait proposer d'éditer le `.bashrc` ou `.profile` pour établir les variables nécessaires à OCAML au démarrage. Vérifiez que l'un des deux fichiers contienne une ligne du genre

```
#OPAM configuration
. $HOME/.opam/opam-init/init.sh > /dev/null 2> /dev/null || true
```

Si ce n'est pas le cas, ajoutez ces lignes au `.profile`

3. Étant donné que `.profile` est lu à l'ouverture de la session, pour charger l'environnement OCAML, il faut ou bien vous déloger et reloger, ou bien charger le `.profile` via `$ source ~/.profile` ou bien évaluer la configuration que propose opam `$ eval $(opam config env)`
4. Pour s'assurer que OPAM est bien détecté, on peut vérifier que la variable d'environnement `PATH` contient un chemin vers OPAM. Pour se faire, il faut vérifier que la sortie de la commande `$ echo $PATH` contienne une entrée contenant un `.opam`.
5. On peut maintenant installer les paquets nécessaires, avec `$ opam install tuareg merlin .`
6. La configuration de l'éditeur peut-être gérée par OPAM avec `$ opam user-setup install .`

2 Compiler du OCAML

La compilation est un processus qui transforme un fichier écrit dans un langage A en un autre fichier écrit dans un langage B . La machine que vous utilisez ne reconnaît qu'un seul langage qui est un langage binaire. Lorsque vous voulez que votre machine exécute du code OCAML, vous allez donc devoir compiler votre fichier écrit en OCAML vers un fichier qui contient du code binaire. Ce programme est appelé `ocamlc`. Il est également possible de compiler de manière un peu plus automatique à l'aide d'un `Makefile`. Réussir à compiler du OCAML est l'objectif de la suite de cette séance.