

Rapport de Bug

N°1 : Erreur de linking : myHash

Florian REYNIER & Mathis CARISTAN

9 mars 2016

1 Contexte

Nous sommes actuellement en train de développer un ensemble de test sur des fonctions de hachage. Pour cela, par commodité nous avons réalisé une interface en OCaml qui appelle des fonctions en C. Ce bug intervient lors de la compilation des sources d'une de nos fonctions en C nommée myHash. Cela nous laisse assez perplexe dans le sens où d'autres fonctions C qui suivent exactement le même schéma d'implémentation comme par exemple de la fonction murmurhash fonctionnent parfaitement. Nous fournissons donc joint à ce rapport les sources en cause (myHash) et celles de murmurhash comme témoin.

2 Implémentation

L'interface Ocaml est exactement la même dans les deux cas l'appel à la fonction C est fait grace au mot clé external (fichiers .ml) la partie C est aussi très semblable une fonction C est dédiée à la communication entre Ocaml et la fonction de hash en elle même, C'est une fonction qui renvoie un "CAMLprim value" et qui est appelée par l'interface Ocaml, les arguments sont passés sous forme d'un tableau "value ml_array" les valeurs sont ensuite récupéré grace aux fonction de la bibliothèque Caml de C incluses de la manière suivante :

```
#include <caml/mlvalues.h>
#include <caml/alloc.h>
#include <caml/memory.h>
```

3 Explication du bug

Lors de l'exécution de la commande "ocamlopt -o myHash myHash.ml myHash.c" où myHash.ml est l'interface Ocaml et myHash.c la fonction.

```
ocamlopt -o myHash myHash.ml myHash.c
```

```
myHash.c: In function `Transform':
```

```
myHash.c:69:9: warning: value computed is not used [-Wunused-value]
```

```
    *mot++;
```

```
    ^
```

```
myHash.c: In function `MyHash':
```

```
myHash.c:77:25: warning: initialization makes pointer from integer without a cast
```

```
    const char** argv = ml_array;
```

```
    ^
```

```
myHash.c:77:18: warning: unused variable `argv' [-Wunused-variable]
```

```
    const char** argv = ml_array;
```

```
    ^
```

```

In file included from myHash.c:7:0:
/usr/lib/ocaml/caml/memory.h:88:29: warning: unused variable 'caml__frame' [-Wunused-variable]
    struct caml__roots_block *caml__frame = caml_local_roots
                                ^
/usr/lib/ocaml/caml/memory.h:91:3: note: in expansion of macro 'CAMLparam0'
    CAMLparam0 (); \
    ^
myHash.c:76:5: note: in expansion of macro 'CAMLparam1'
    CAMLparam1 ( ml_array );
    ^
myHash.o: dans la fonction « MyHash »:
myHash.c:(.text+0x2d8): définitions multiples de « MyHash »
myHash.o:myHash.c:(.text+0x2d8): défini pour la première fois ici
/tmp/camlstartup46a774.o: dans la fonction « caml_program »:
(.text+0x15): référence indéfinie vers « camlMyHash__entry »
/tmp/camlstartup46a774.o: dans la fonction « caml_globals »:
(.data+0x2c0): référence indéfinie vers « camlMyHash »
/tmp/camlstartup46a774.o: dans la fonction « caml_data_segments »:
(.data+0x3b0): référence indéfinie vers « camlMyHash__data_begin »
/tmp/camlstartup46a774.o: dans la fonction « caml_data_segments »:
(.data+0x3b8): référence indéfinie vers « camlMyHash__data_end »
/tmp/camlstartup46a774.o: dans la fonction « caml_code_segments »:
(.data+0x3f8): référence indéfinie vers « camlMyHash__code_begin »
/tmp/camlstartup46a774.o: dans la fonction « caml_code_segments »:
(.data+0x400): référence indéfinie vers « camlMyHash__code_end »
/tmp/camlstartup46a774.o: dans la fonction « caml_frametable »:
(.data+0x438): référence indéfinie vers « camlMyHash__frametable »
collect2: error: ld returned 1 exit status
File "caml_startup", line 1:
Error: Error during linking

```

4 Explication complémentaire

Aux vues du retour du compilateur, il semblerait que les includes ne fonctionnent pas, impression renforcée du fait qu'en enlevant ces includes le message d'erreur reste exactement le même. Nous n'avons pour le moment aucune idée de pourquoi ces includes seraient ignorés dans ce cas précis.