Report about a memory leak error on **ocaml-mad** and **ocaml-vorbis**

Sumin Ahn
GIST PSL

March 08, 2016

1 OCaml/C programs reported by Furr

Michael Furr et al. [1] reported that **ocaml-mad-0.1.0** has possibility of leaking memory or causing subtle memory corruption. The error comes from registering a local parameter with the garbage collecter but then forgetting to release it.

1.1 ocaml-mad

ocaml-mad is a high-quality MPEG audio decoder implemented with OCaml.
we checked the error by executing modules in **ocaml-mad-0.1.0** with Memcheck in Valgrind. Fortunately, there is an example from the developer of **ocaml-mad**, which is named **mp32wav**. **mp32wav** is an mp3 to wav converter using **ocaml-mad**. It calls most of modules in **ocaml-mad-0.1.0**. And the result executing it with Memcheck is on Figure 1a.
The result shows that 1,536.048 bytes in 3 blocks are possibly lost by copy_buffer in **mad_stub.c**(See line 7), which contains mad-library modules to be used by **mad.ml**.

1.2 ocaml-vorbis

ocaml -vorbis is OCaml bindings for the libvorbis, which is an open-source audio codec that uses **.ogg** extention. we also checked the example program **ogg2wav ocaml-vorbis-0.2.0** with it. See Figure 1b. Valgrind reported that Vorbis.get_comments makes memory leaks (line 8). This function calls ocaml_vorbis_utf8_decode in **vorbis_stubs.c** (line 7).

2 main cause

Figure 2 shows copy_buffer in **mad_stub.c** and ocaml_vorbis_utf8_decode in **vorbis_stub.c**. Let's start with **ocaml-mad**. It uses CAMLlocal1 to declare res(line

3). Then returned res directly but didn't use CAMLreturn macro(line 6). CAMLreturn macro is to tell the GC that we've finished with those local variables declared by CAMLlocal. Thus plain return can cause a memory leak in this case.
The problem in **ocaml-vorbis** is very similar to the previous one. in ocaml_vorbis_utf8_decode, char* utf8 was declared as C local variable but returned by CAMLreturn.

3 result

Figure 3 shows memory usage by time for **ocaml-mad-0.1.0** and **0.1.1**. In both graph we repeatedly executed copy_buffer. Before fixing the memory leak, in **ocaml-mad-0.1.0**, memory used by mp32wav increases linearly as time goes. On the other hand, after fixing, it remains constant as 3204kb.

Figure 4 shows memory usage by time for **ocaml-vorbis-2.1.0**. We also can check the linear increase of the memory usage.

References

[1] Michael Furr and Jeffrey S. Foster. Checking type safety of foreign function calls. In *Proceedings of the 2005 ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '05, pages 62–72, New York, NY, USA, 2005. ACM.

```
1. 1,536,048 bytes in 3 blocks are possibly lost in loss
   record 33 of 33
2.   at 0x402A17C: malloc (in
   /usr/lib/valgrind/vgpreload_memcheck-x86-linux.so)
3.   by 0x8064959: caml_aligned_malloc (in
   /home/s20125047/Downloads/ocaml-mad/examples/mp32wav)
4.   by 0x805338E: caml_alloc_for_heap (in
   /home/s20125047/Downloads/ocaml-mad/examples/mp32wav)
5.   by 0x8053570: caml_alloc_shr (in
   /home/s20125047/Downloads/ocaml-mad/examples/mp32wav)
6.   by 0x8053AAB: caml_alloc_string (in
   /home/s20125047/Downloads/ocaml-mad/examples/mp32wav)
7.   by 0x80507BC: copy_buffer (mad_stubs.c:46)
8.   by 0x80507BC: ocaml_decode (mad_stubs.c:296)
9.   by 0x8064042: caml_interprete (in
   /home/s20125047/Downloads/ocaml-mad/examples/mp32wav)
10.    by 0x8051451: caml_main (in
   /home/s20125047/Downloads/ocaml-mad/examples/mp32wav)
11.    by 0x8062E66: main (in
   /home/s20125047/Downloads/ocaml-mad/examples/mp32wav)
```

Figure 1a: ocaml-mad

```
1    360,760 bytes in 6,220 blocks are definitely lost in
   loss record 41 of 43
2.   at 0x402C324: realloc (in
   /usr/lib/valgrind/vgpreload_memcheck-x86-linux.so)
3.   by 0x8070033: charset_convert (charset.c:512)
4.   by 0x806F454: convert_buffer (utf8.c:262)
5.   by 0x806F454: convert_string (utf8.c:278)
6.   by 0x806F5F0: utf8_decode (utf8.c:318)
7.   by 0x806EEBE: ocaml_vorbis_utf8_decode
   (vorbis_stubs.c:527)
8.   by 0x804F535: camlVorbis__get_comments_1137 (in
   /home/s20125047/Downloads/vorbis/ocaml-vorbis-0.2.0/
examples/ogg2wav)
9.   by 0x804CE0A: camlOgg2wav__entry (in
   /home/s20125047/Downloads/vorbis/ocaml-vorbis-0.2.0/
examples/ogg2wav)
10.  by 0x804B4DC: caml_program (in
   /home/s20125047/Downloads/vorbis/ocaml-vorbis-0.2.0/
examples/ogg2wav)
11.  by 0x8086BC5: ??? (in
   /home/s20125047/Downloads/vorbis/ocaml-vorbis-0.2.0/
examples/ogg2wav)
12.  by 0x8076479: caml_main (in
   /home/s20125047/Downloads/vorbis/ocaml-vorbis-0.2.0/
examples/ogg2wav)
13.  by 0x80764BB: main (in
   /home/s20125047/Downloads/vorbis/ocaml-vorbis-0.2.0/
examples/ogg2wav)
```

Figure 1b: ocaml-vorbis

ocaml-mad

```
1. static value copy_buffer(char const *b, int len)
2. {
3.  CAMLlocal1(res);
4.  res = alloc_string(len);
5.  memmove(String_val(res), b, len);
6.  return res;
7. }
```

ocaml-vorbis

```
1. CAMLprim value ocaml_vorbis_utf8_decode(value string)
2. {
3.  CAMLparam1(string);
4.      char* utf8;
5.  if(utf8_decode(String_val(string),&utf8) >= 0)
6.    {
7.     CAMLreturn(caml_copy_string(utf8));
8.    }
9.     caml_raise_with_arg(
        *caml_named_value("vorbis_exn_utf8_failure"),
           string);
10.  }
```

Figure 2: Main cause

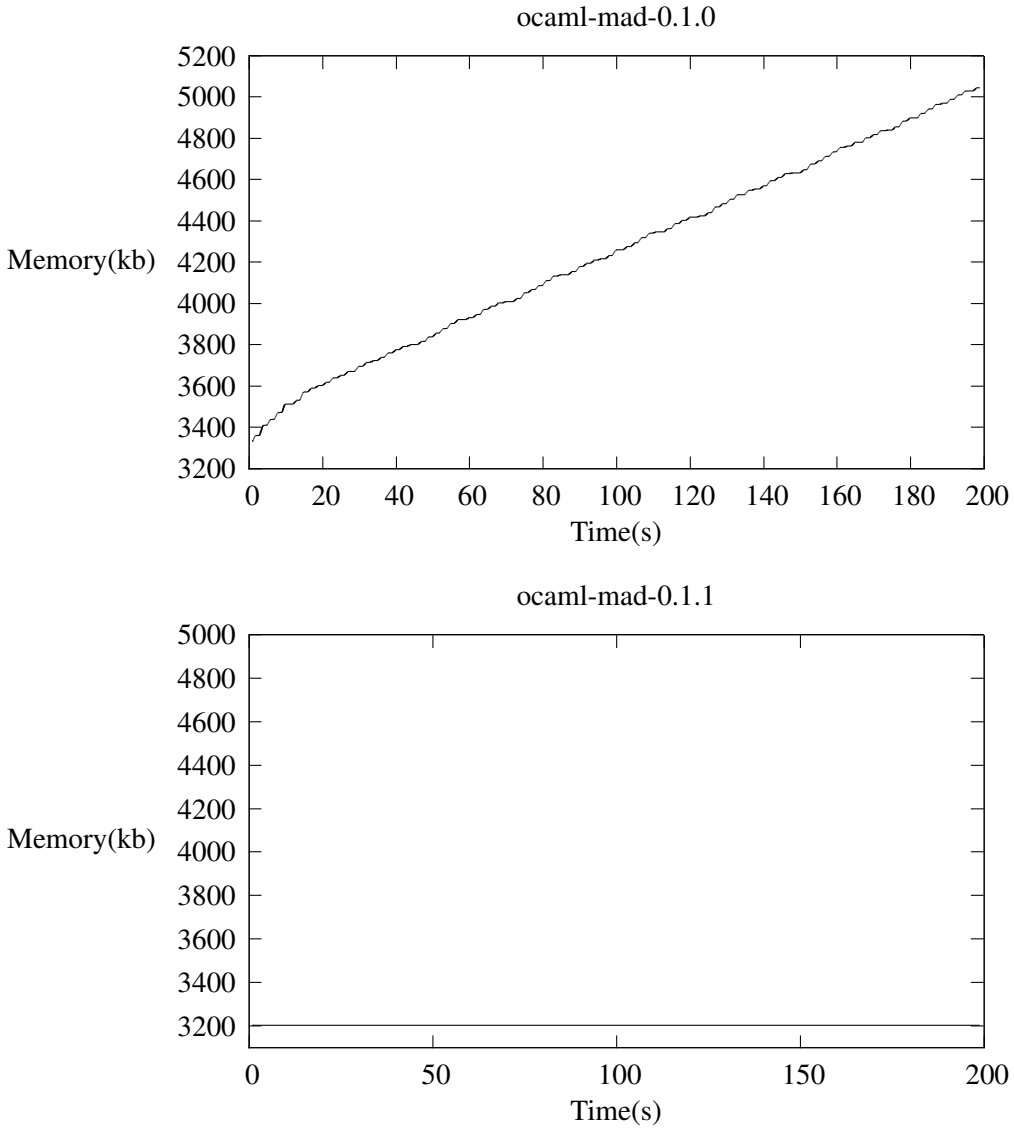

Figure 3: mad-memory usage

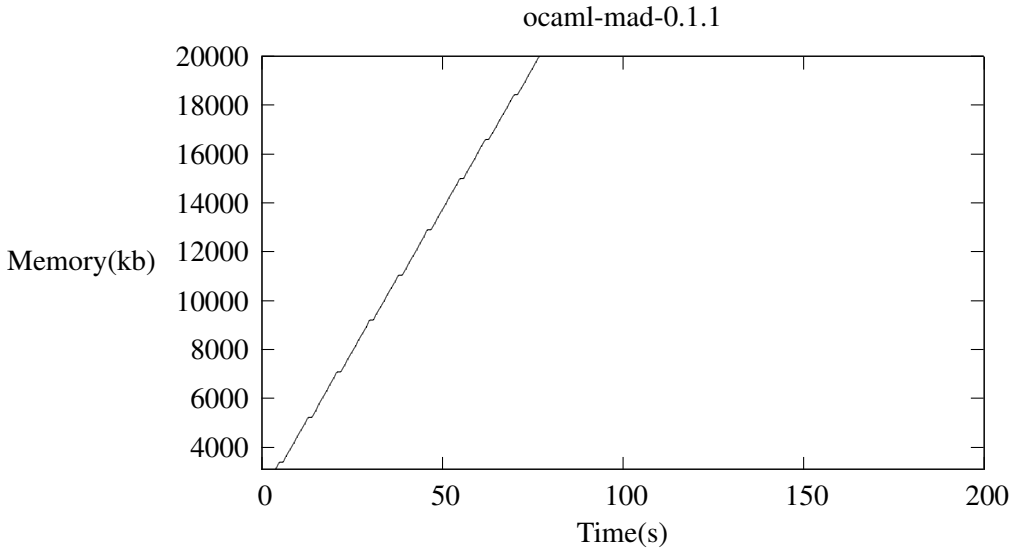

Figure 4: vorbis-memory usage