Subject: MemorySanitizer: use-of-uninitialized-value in CoWork
Posted by Novo on Fri, 19 Jul 2019 20:17:48 GMT
View Forum Message <> Reply to Message

Could you please apply a patch below to CoWork?
MemorySanitizer is complaining about uninitialized memory. Theoretically, default constructor of std::exception_ptr is supposed to initialized it, but practically I'm getting a warning.
The problem can be easily fixed.

```
diff --git a/uppsrc/Core/CoWork.cpp b/uppsrc/Core/CoWork.cpp
index 46e49dc15..abfbae2b8 100644
--- a/uppsrc/Core/CoWork.cpp
+++ b/uppsrc/Core/CoWork.cpp
@@ -113,7 +113,7 @@ void CoWork::Pool::DoJob(MJob& job)
        }

        lock.Leave();
-       std::exception_ptr exc;
+       std::exception_ptr exc = nullptr;
        try {
                if(looper)
                        work->looper_fn();
@@ -370,6 +370,7 @@ int CoWork::GetWorkerIndex()
 }

 CoWork::CoWork()
+: exc(nullptr)
 {
        LLOG("CoWork constructed " << FormatHex(this));
        todo = 0;
```

Patched file is attached.

File Attachments

1) CoWork.cpp, downloaded 257 times

Subject: Re: MemorySanitizer: use-of-uninitialized-value in CoWork
Posted by mirek on Fri, 19 Jul 2019 21:50:42 GMT
View Forum Message <> Reply to Message

This really feels like sanitizer bug, but whatever....

Subject: Re: MemorySanitizer: use-of-uninitialized-value in CoWork
Posted by Novo on Fri, 19 Jul 2019 23:27:28 GMT
View Forum Message <> Reply to Message

mirek wrote on Fri, 19 July 2019 17:50This really feels like sanitizer bug, but whatever....
Thanks!

I do not think this is a bug. A team working on sanitizers is exceptionally good.
MemorySanitizer reports all uninitialized memory reads, including, for example, memmove and memcpy of uninitialized memory. This is not necessarily a bug, but this allows the sanitizer to be fast unlike valgrind.
The problem is that all sanitizers exit app on first detected error.

Subject: Re: MemorySanitizer: use-of-uninitialized-value in CoWork
Posted by mirek on Sat, 20 Jul 2019 06:56:25 GMT
View Forum Message <> Reply to Message

Novo wrote on Sat, 20 July 2019 01:27mirek wrote on Fri, 19 July 2019 17:50This really feels like sanitizer bug, but whatever....
Thanks!

I do not think this is a bug. A team working on sanitizers is exceptionally good.
MemorySanitizer reports all uninitialized memory reads, including, for example, memmove and memcpy of uninitialized memory. This is not necessarily a bug, but this allows the sanitizer to be fast unlike valgrind.
The problem is that all sanitizers exit app on first detected error.

Well, but obviously we are not doing anything bad here. So it must be either a bug in sanitizer or in standard library (or perhaps in compiler).

Mirek
