Subject: ARM threadind does not work
Posted by Didier on Sat, 15 Dec 2012 10:00:01 GMT
View Forum Message <> Reply to Message

Hi all,

When compiling apps whith 'MT' falg set, the applications get stuck on a mutex:
#1 0x400bd138 in pthread_mutex_lock () from /lib/libpthread.so.0
#2 0x001295e4 in Upp::Mutex::Enter (this=0x50803d) at /users/didier/upp/uppsrc/Core/Mt.h:305
#3 0x00129624 in Upp::Mutex::Lock::Lock (this=0xbeed4b0c, s=...) at
/users/didier/upp/uppsrc/Core/Mt.h:353
u#4 0x002a23f4 in Upp::StaticMutex::Initialize (this=0x506118) at
/users/didier/upp/uppsrc/Core/Mt.cpp:643
#5 0x00129550 in Upp::StaticMutex::Get (this=0x506118) at
/users/didier/upp/uppsrc/Core/Mt.h:380
#6 0x00129580 in Upp::StaticMutex::operator Upp::Mutex& (this=0x506118) at
/users/didier/upp/uppsrc/Core/Mt.h:381
#7 0x002e0800 in Upp::AddModule (l=0x3ffa50, name=0x0) at
/users/didier/upp/uppsrc/Core/t.cpp:98
#8 0x0000fbb0 in Upp::LNG_MODULE00_fn () at /users/didier/upp/uppsrc/Core/t.h:211
#9 0x0000fc2c in Upp::Callinit::Callinit (this=0x4141dd, fn=0xfb98 <Upp::LNG_MODULE00_fn()>,
cpp=0x3479e8 "/users/didier/upp/uppsrc/Core/t.h", line=202)
    at /users/didier/upp/uppsrc/Core/Defs.h:169
#10 0x0000fb18 in __static_initialization_and_destruction_0 (__initialize_p=1, __priority=65535)
at /users/didier/upp/uppsrc/Core/t.h:202
#11 0x0000fb70 in global constructors keyed to CtrlLib.icpp(void) () at
/users/didier/upp/uppsrc/CtrlLib/CtrlLib.icpp:20
#12 0x003474c0 in __libc_csu_init ()
#13 0x4043efb8 in __libc_start_main () from /lib/libc.so.6
#14 0x0000e338 in _start ()

I will be looking into it this WE but if someone has any idea I would appreciate it.

Apparently it get's stuck when taking the mutex on function start:
```
void AddModule(const LngEntry__* l, const char *name)
{
 CriticalSection::Lock __(slng);   ********** HERE **********
 Array<LngModule>& ma = sMod();
 LngModule& m = ma.Add();
 m.name = name;
 Vector<LngRec> *lr = NULL;
 while(l->lang) {
  if(l->lang == 1) {
   CharS ids;
   ids.s = l->text;
   lr = &m.map.GetAdd(ids);
   LngRec& r = lr->Add();
```

```
   r.lang = LNG_('E','N','U','S');
   r.text = GetENUSc(l->text);
  }
  else
  if(l->text && *l->text) {
   LngRec& r = lr->Add();
   r.lang = l->lang;
   r.text = l->text;
  }
  l++;
 }
}
```

Subject: Re: ARM threadind does not work
Posted by Didier on Sat, 15 Dec 2012 17:16:04 GMT
View Forum Message <> Reply to Message

Found one uninitialized variable:

Core/Mt.cpp L13 :

```
static Mutex& sMutexLock()
{
 static Mutex *section;   **** need's to be initialized to 0  ****
 if(!section) {
  static byte b[sizeof(Mutex)];
  section = new(b) Mutex;
 }
 return *section;
}
```

If section is not initialized to '0' : we can pray for the lazy initialisation to work properly

This correction, is yet not sufficient

NB: I will furnish a complete patch when I'll get this running (if I succeed)

Subject: Re: ARM threadind does not work
Posted by Didier on Sat, 15 Dec 2012 21:20:57 GMT
View Forum Message <> Reply to Message

Ok, I found a correction,

The correction only shows where the problem is: incorrect management of 'sMutexLock' initialization by several threads at the same time
A clean correction is still to be found : I think a refactoring of StaticMutex will be necessary for this (the correction is only valid for POSIX systems)

This is not the first time Upp stumbles on the same problem : there is another case stated on the forum:
http://www.ultimatepp.org/forum/index.php?t=msg&goto=15998&&srch=sMutexLock#msg_15998

Core/Mt.cpp L14

BEFORE:
```
static Mutex& sMutexLock()
{
 static Mutex *section=0;
 if(!section) {
  static byte b[sizeof(Mutex)];
  section = new(b) Mutex;
 }
 return *section;
}
```

AFTER
```
static Mutex *_sMutexLock_section=0;
static void _sMutexLock()
{
 if(!_sMutexLock_section) {
  static byte b[sizeof(Mutex)];
  _sMutexLock_section = new(b) Mutex;
 }
}

static Mutex& sMutexLock()
{
 static pthread_once_t sMutexLock_is_initialized = PTHREAD_ONCE_INIT;
    (void) pthread_once(&sMutexLock_is_initialized, _sMutexLock);
 return *_sMutexLock_section;
}
```

Subject: Re: ARM threadind does not work
Posted by Didier on Tue, 18 Dec 2012 20:46:03 GMT
View Forum Message <> Reply to Message

Hoooo !

I forgot to precise:

image.

But this bug will probably appear from time to time on other architectures / machines

---

Subject: Re: ARM threadind does not work
Posted by mirek on Fri, 28 Dec 2012 10:59:15 GMT
View Forum Message <> Reply to Message

This is all somewhat weird.

```
static Mutex& sMutexLock()
{
 static Mutex *section;
 if(!section) {
  static byte b[sizeof(Mutex)];
  section = new(b) Mutex;
 }
 return *section;
}

INITBLOCK {
 sMutexLock();
}
```

First of all, static variables are by definition initialized to zero. Putting " = 0" IMO can result in creating race condition, as the setting this explicit zero might be postponed to the time sMutexLock is called for the first time.

Anyway, all of that is sort of irrelevant: As you can see, INITBLOCK should ensure that sMutexLock is called at least once before APP_MAIN starts, ergo it should be already initialized when the first thread has chance screw things up.

I would recommend placing some LOGs into the INITBLOCK, then into sMutexLock and at the start of APP_MAIN to find out what is really going on... (it is quite possible that INITBLOCK does not work as we wish, in that case it would be very good to know that in order to find some workarounds, as we depend on it a lot)

Mirek

Subject: Re: ARM threadind does not work
Posted by Didier on Fri, 28 Dec 2012 13:40:24 GMT
View Forum Message <> Reply to Message

Hy Mirek,

Quote:First of all, static variables are by definition initialized to zero.

While this is true when compiling in debug mode, it is not true with all compilers when compiled with O2 or O3 so I'm used to not counting on the default init of variables

Quote: IMO can result in creating race condition, as the setting this explicit zero might be postponed to the time sMutexLock is called for the first time.
For me static vars are initialised a exe startup (before app starts) and if there could be a race condition then the code can not work properlly (see pthread man and the reason of pthread_once_t existence)

Quote:Anyway, all of that is sort of irrelevant: As you can see, INITBLOCK should ensure that sMutexLock is called at least once before APP_MAIN starts, ergo it should be already initialized when the first thread has chance screw things up.
That's what i also thought when first ran into the bug (so when I saw the inititalisation problem, I thought I had solutionned the problem ... but that was not the case), but after some reading and trying pthread_once_t the issue was corrected.

I agree that this bug has to find a valid correction (my pacth only deals with one use of StaticMutex ... there are others.

I didn't try to debug the internals for now (lack of time).
I'll try to do it next week.

Subject: Re: ARM threadind does not work
Posted by Novo on Mon, 31 Dec 2012 06:16:48 GMT
View Forum Message <> Reply to Message

Didier wrote on Fri, 28 December 2012 08:40Hy Mirek,

Quote:First of all, static variables are by definition initialized to zero.

While this is true when compiling in debug mode, it is not true with all compilers when compiled with O2 or O3 so I'm used to not counting on the default init of variables

C++ Standard.
3.6.2 Initialization of non-local objects

1 Objects with static storage duration (3.7.1) shall be zero-initialized (8.5) before any other initialization takes place.

Subject: Re: ARM threadind does not work
Posted by Didier on Mon, 31 Dec 2012 11:09:50 GMT
View Forum Message <> Reply to Message

Hi Novo,

Quote:C++ Standard.
3.6.2 Initialization of non-local objects
1 Objects with static storage duration (3.7.1) shall be zero-initialized (8.5) before any other initialization takes place.

OK, I didn't know that, thank's
At work we often use C (not C++) for embedded products, and we run into this problem every day.

Subject: Re: ARM threadind does not work
Posted by Zbych on Mon, 31 Dec 2012 11:42:59 GMT
View Forum Message <> Reply to Message

Even plain old ANSI C requires initialization of static variables, so I guess you are another satisfied user of Microchip PIC compilers?

Subject: Re: ARM threadind does not work
Posted by Didier on Tue, 01 Jan 2013 09:55:34 GMT
View Forum Message <> Reply to Message

Quote:Even plain old ANSI C requires initialization of static variables, so I guess you are another satisfied user of Microchip PIC compilers?

Nop, not at all.
I work with big systems using different OS Linux/VxWorks C/C++ GCC but what I did not say clearly enough is that initialising vars is a good habit (whether they are static or not).