

Re: pthread_mutex_unlock (was Re: sched_yield() makes OpenLDAP slow)

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-01/msg10144.html>

- *From:* Steven Rostedt <rostedt@xxxxxxxxxxxx>
 - *Date:* Thu, 26 Jan 2006 23:27:50 -0500
-

On Thu, 2006-01-26 at 13:56 -0800, Howard Chu wrote:

> Nick Piggin wrote:

>>>

>>> But why does A take the mutex in the first place? Presumably because
>>> it is about to execute a critical section. And also presumably, A
>>> will not release the mutex until it no longer has anything critical
>>> to do; certainly it could hold it longer if it needed to.

>>>

>>> If A still needed the mutex, why release it and reacquire it, why not
>>> just hold onto it? The fact that it is being released is significant.

>>>

>>

>> Regardless of why, that is just the simplest scenario I could think
>> of that would give us a test case. However...

>>

>> Why not hold onto it? We sometimes do this in the kernel if we need
>> to take a lock that is incompatible with the lock already being held,
>> or if we discover we need to take a mutex which nests outside our
>> currently held lock in other paths. Ie to prevent deadlock.

>

> In those cases, A cannot retake the mutex anyway. I.e., you just said
> that you released the first mutex because you want to acquire a
> different one. So those cases don't fit this example very well.

Lets say you have two locks X and Y. Y nests inside of X. To do block1
you need to have lock Y and to do block2 you need to have both locks X
and Y, and block 1 must be done first without holding lock X.

```
func()
{
again:
mutex_lock(Y);
block1();
if (!mutex_try_lock(X)) {
mutex_unlock(Y);
mutex_lock(X);
mutex_lock(Y);
}
```

Re: pthread_mutex_unlock (was Re: sched_yield() makes OpenLDAP slow)

```
if (block1_has_changed()) {
mutex_unlock(Y);
mutex_unlock(X);
goto again;
}
}
block2();
mutex_unlock(X);
mutex_unlock(Y);
}
```

Stuff like the above actually is done (it's done in the kernel). So you can see here that Y can be released and reacquired right away. If another task was waiting on Y (of lower priority) we don't want to give up the lock, since we would then block and the chances of block1_has_changed goes up even more.

>
>> Another reason might be because we will be running for a very long
>> time without requiring the lock.
>
> And again in this case, A should not be immediately reacquiring the lock
> if it doesn't actually need it.

I'm not sure what Nick means here, but I'm sure he didn't mean it to come out that way ;)

>
>> Or we might like to release it because
>> we expect a higher priority process to take it.
>
> And in this case, the expected behavior is the same as I've been pursuing.

But you can't know if a higher or lower priority process is waiting. Sure it works like what you say when a higher priority process is waiting, but it doesn't when it's a lower priority process waiting.

— Steve

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>

• **References:**

- ◆ ***RE: pthread_mutex_unlock (was Re: sched_yield() makes OpenLDAP slow)***
◇ From: David Schwartz

Re: pthread_mutex_unlock (was Re: sched_yield() makes OpenLDAP slow)

Re: pthread_mutex_unlock (was Re: sched_yield() makes OpenLDAP slow)

- ◆ [Re: pthread_mutex_unlock \(was Re: sched_yield\(\) makes OpenLDAP slow\)](#)
 ◇ From: Howard Chu
- ◆ [Re: pthread_mutex_unlock \(was Re: sched_yield\(\) makes OpenLDAP slow\)](#)
 ◇ From: Nick Piggin
- ◆ [Re: pthread_mutex_unlock \(was Re: sched_yield\(\) makes OpenLDAP slow\)](#)
 ◇ From: Howard Chu
- ◆ [Re: pthread_mutex_unlock \(was Re: sched_yield\(\) makes OpenLDAP slow\)](#)
 ◇ From: Nick Piggin
- ◆ [Re: pthread_mutex_unlock \(was Re: sched_yield\(\) makes OpenLDAP slow\)](#)
 ◇ From: Howard Chu

- Prev by Date: [Re: \[00/23\] \[Suspend2\] Freezer Upgrade Patches](#)
- Next by Date: [Re: Problems with MSI-X on ia64](#)
- Previous by thread: [Re: pthread_mutex_unlock \(was Re: sched_yield\(\) makes OpenLDAP slow\)](#)
- Next by thread: [Re: pthread_mutex_unlock \(was Re: sched_yield\(\) makes OpenLDAP slow\)](#)
- Index(es):
 - ◆ [Date](#)
 - ◆ [Thread](#)