

Re: Kamaelia 0.4.0 RELEASED – Faster! More Tools! More Examples! More Docs! ;-)

Re: Kamaelia 0.4.0 RELEASED – Faster! More Tools! More Examples! More Docs! ;-)

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2006-06/msg03385.html>

- *From:* Michael <ms@xxxxxxxxxxxxx>
 - *Date:* Sat, 24 Jun 2006 18:30:06 +0100
-

Robin Becker wrote:

Michael wrote:

I'm extremely pleased to say – Kamaelia 0.4.0 has been released!

Windows users might see problems with the attempt at importing libc.so.6 in the Axon Scheduler.

I changed the lines at line 44 of scheduler.py to something like this

```
[ .. patch .. ]
```

Many thanks!

Yes, this issue has also been raised on the Kamaelia mailing list in the past couple of days by Stefan Rank (he had a solution based on using win32.Sleep(0,0) which calls SleepEx[1]). I'm planning on an Axon 1.5.1 release shortly to deal with this issue.

[1]

<http://www-128.ibm.com/developerworks/eserver/library/es-MigratingWin32toLinux.html?ca=dgr-lnxw09CPP2Linu>

I've got three options – I can either take the approach you've suggested, the approach suggested on the list (which is to call win32.Sleep(0,0), or actually remove this code from the scheduler completely.

I'm actually most tempted by the final option, largely because this code was added in an earlier release as a way of making Axon & the scheduler more system friendly. Since we've now effectively gone from a busy wait loop to a scheduler that /can/ sleep waiting for threads [2] this means that code can be a lot more system friendly by default.

[2] For example, rather than calling select with a timeout of zero in the main thread, it's now called with a timeout of 5s in a thread, which wakes up the scheduler when there's something to do. The pygame code does the same trick...

That said, there is still whole chunks of code that busy run (such as the

Re: Kamaelia 0.4.0 RELEASED – Faster! More Tools! More Examples! More Docs! ;-)

Re: Kamaelia 0.4.0 RELEASED – Faster! More Tools! More Examples! More Docs! ;-)

physics simulation, so I'm torn. Do I take the code you put forward, the solution put forward on the the list which I can't really check (since I don't have a windows box to test with right now (I will on Monday though)), or yank it out.

I suppose the "correct" thing is to put your solution into Axon 1.5.1 – to get rid of the failure, and then to test Stefan's solution locally. My hesitation with regard to Stefan's solution is simply the fact that I don't tend to do a huge amount of coding under windows, and hence have greater concern there as a result.

I'll get a release out with that fix either today or tomorrow, and then roll up Stefan's more detailed fix into the 1.5.2 release along with anything else that crops up over the next few days (hopefully nothing, but you never know).

This allows somethings to run eg

```
AxonVisualiser.py --navelgaze
```

but I'm not sure if the results are really unaffected by not having a real yielder. The diagram appears, but doesn't seem to settle down.

I don't think the AxonVisualiser would be particularly affected by this – any wobbling you'll see there is probably more down to the simple physics model the visualiser uses not having sufficient dampening for the layout you see. (The physics model was apparently originally written by Matt for a lava lamp simulator, but has turned out to be incredibly useful as a simple layout tool :-)

A little bit of manual assistance to spread things out can sometimes help it settle down though :)

whoops, forgot to say thanks as I think this is very nice work

You're very welcome. If you have any other suggestions/ideas for improvement they'd be very welcome :-)

Personally, I think some of the most fun stuff right now being done by new people with it is coming from 2 of our summer of code students. One, Thomas, is working on 3D components (eg putting pygame components into a PyOpenGL display).

The other is doing bittorrent related work, but has also written a webserver and IRC bot – both pretty much from scratch (at the protocol level) over the past month. (The other two students are working on encryption ideas for tools for communications – eg encrypted VoIP type things)

Re: Kamaelia 0.4.0 RELEASED – Faster! More Tools! More Examples! More Docs! ;-)

Re: Kamaelia 0.4.0 RELEASED – Faster! More Tools! More Examples! More Docs! ;-)

Thomas's work (3D) is discussed here: <http://thfsoc.blogspot.com/>
Ryan's code (Bit Torrent, webserver/client, IRC bot) is sitting here:
* <http://kamaelia.cvs.sourceforge.net/kamaelia/Sketches/RJL/>

Thanks for the bug fix/kind words!

Regards,

Michael.