

README

Perry Kundert

March 29, 2016

Contents

1	Thread-safe Python schedulers	1
1.1	Deficiencies	1
1.1.1	Python 2/3 Consistency	1
1.1.2	Thread safety	2
1.1.3	Ordering	2

1 Thread-safe Python schedulers

1.1 Deficiencies

The sched.scheduler implementation supplied with Python omits some capabilities that are desirable:

1.1.1 Python 2/3 Consistency

An identical API should be available in Python 2, so that applications are consistent between and compatible with Python 2 and 3. This requires an implementation of sched.scheduler in Python 2 that:

- Has ‘kwargs’ Option

When an event is scheduled with ‘enter’ or ‘enterabs’, a ‘kwargs’ keyword needs to be saved. In the Python 3 version, the sched.Event ‘namedtuple’ is defined to contain these both the position ‘argument’ list and keywords ‘kwargs’; the Python 2 version sched.Event only has the ‘argument’ list.

Therefore, in the Python 2 version, we'll prepare a closure to capture the supplied 'kwargs', and pass them when invoking the supplied 'action', when invoking the base-class 'enterabs'.

1.1.2 Thread safety

Scheduling new events and/or removing existing events from other threads isn't always correct. For example, the run method will not awaken and re-schedule itself, if an event is added before the previous earliest event's expiry.

1.1.3 Ordering

The ordering is strictly by timeout, then priority. Often, other ordering schemes are desirable. For example, ordering by timeout, but then scaling event age by priority to avoid starvation of lower-priority events.