Потоки: неправильные приемы работы

Андрей Светлов

asvetlov.blogspot.com

Потоки – зло!

Модуль thread — зло.

Используйте threading.

threading.Lock — тоже зло.

Используйте threading.RLock

Контекстный менеджер — хорошо

```
with self.mutex:
    Do()
```

try .. finally тоже неплохо, хоть и длинно

```
try:
    self.mutex.acquire()
    do()
finally:
    self.mutex.release()
```

Все остальное — зло. Исключения — непредсказуемы.

Трижды подумайте, прежде чем использовать threading.Event или threading.Semaphore

Полезней работать в терминах threading.Condition

Запуск потока: наследование от threading.Thread не совсем хорошо.

Иерархия классов должна быть естественной.

«Как мне прибить поток?»
Это — преступление.

К счастью, нереализуемое.

Ресурсы нужно освобождать.

Всегда используйте threading.Thread.join

```
import threading

def f():
    while True:
        pass

th = threading.Thread(target=f)
th.start()
th.join()
```

Потоки-демоны, как правило — зло.

Создаются неумелыми программистами, не знающими о .join()

```
th.setDaemon(True)
```

Во время завершения интерпретатора демоны еще работают...

Поток завершается посылкой ему сигнала.

- threading.Condition
- select.select
- select.poll

Ждать вечно — обычно плохая идея.

Правильнее ждать с большим таймаутом.
Внимание: у threading.Condition.wait таймаут есть!

Вопросы?

Андрей Светлов

asvetlov.blogspot.com