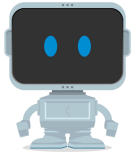
Про процессы, и потоки, и Python

Стас Рудаков
stas@garage22.net

Minsk Python Meetup
August 2018



DataRobot

```python
import logging
from multiprocessing import Process
from threading import Thread
import time

logging.basicConfig(level='INFO')

def foobar(identity):
    for i in range(1000):
        logging.info('I am {}'.format(identity))
        time.sleep(0.001)

t = Thread(target=foobar, args=('thread',))
p = Process(target=foobar, args=('process',))

t.start()
p.start()

t.join()
logging.info('thread joined')
p.join()
logging.info('process joined')
```

Запустим

```
python3 /src/example.py
```

```
[...]
INFO:root:I am thread
INFO:root:I am process
INFO:root:I am thread
INFO:root:I am process
INFO:root:I am thread
INFO:root:I am process
INFO:root:I am thread
INFO:root:I am process
INFO:root:I am thread
INFO:root:I am process
INFO:root:thread joined
INFO:root:I am process
INFO:root:I am process
INFO:root:I am process
INFO:root:I am process
INFO:root:process joined
```

Запустим еще несколько раз... упс

```
for i in $(seq 100); do echo attempt $i; python3 /src/example.py; done
```

```
[...]
attempt 12
[...]
INFO:root:I am thread
INFO:root:I am thread
INFO:root:I am thread
INFO:root:I am thread
INFO:root:I am thread
INFO:root:I am thread
INFO:root:I am thread
INFO:root:I am thread
INFO:root:I am thread
INFO:root:I am thread
INFO:root:I am thread
INFO:root:thread joined
```

Для истории: подготовка окружения

```
$ docker run -it -v `pwd`:/src --privileged ubuntu:18.04
root@06f40dc80b14:/# apt update
[...]
root@06f40dc80b14:/# apt install python3
[...]
```

Что вообще происходит?

```
root@e681a8ffef5b:/# ps -ef --forest
UID        PID  PPID  C STIME TTY          TIME CMD
root       626     0  0 05:49 pts/1    00:00:00 bash
root       638   626  0 05:49 pts/1    00:00:00  \_ ps -ef --forest
root         1     0  0 Aug15 pts/0    00:00:00 /bin/bash
root       623     1  0 05:43 pts/0    00:00:00 python3 /src/example.py
root       625   623  0 05:43 pts/0    00:00:00  \_ python3 /src/example.py
```

```python
import logging
from multiprocessing import Process
from threading import Thread
import time

logging.basicConfig(level='INFO')

def foobar(identity):
    for i in range(1000):
        logging.info('I am {}'.format(identity))
        time.sleep(0.001)

t = Thread(target=foobar, args=('thread',))
p = Process(target=foobar, args=('process',))

t.start()
p.start()

t.join()
logging.info('thread joined')
p.join()
logging.info('process joined')
```

Нам нужен дебаггер!

```
root@06f40dc80b14:/# apt install gdb
[...]
```

Подебажим

```
gdb -p 625
```

```
[...]
For help, type "help".
Type "apropos word" to search for commands related to "word".
Attaching to process 625
Reading symbols from /usr/bin/python3.6...(no debugging symbols found)
    ...done.
[...]
(gdb)
```

```
(gdb) bt
#0  0x00007f0cbc6a66d6 in futex_abstimed_wait_cancelable (private=0,
    abstime=0x0, expected=0, futex_word=0x15b82e0)
    at ../sysdeps/unix/sysv/linux/futex-internal.h:205
#1  do_futex_wait (sem=sem@entry=0x15b82e0, abstime=0x0)
    at sem_waitcommon.c:111
#2  0x00007f0cbc6a67c8 in __new_sem_wait_slow (sem=0x15b82e0, abstime=0
    x0)
    at sem_waitcommon.c:181
#3  0x000000000043f0a8 in PyThread_acquire_lock_timed ()
#4  0x000000000058fbfd in ?? ()
#5  0x00000000004c549b in _PyCFunction_FastCallKeywords ()
#6  0x000000000054ffe4 in ?? ()
#7  0x00000000005546cf in _PyEval_EvalFrameDefault ()
#8  0x000000000054f0e8 in ?? ()
#9  0x0000000000550116 in ?? ()
#10 0x00000000005546cf in _PyEval_EvalFrameDefault ()
#11 0x000000000054f0e8 in ?? ()
#12 0x0000000000550116 in ?? ()
---Type <return> to continue, or q <return> to quit---
```

Символизируем

```
root@06f40dc80b14:/# apt install python3-dbg
[...]
```

Подебажим с символами

```
gdb -p 625
```

```
[...]
For help, type "help".
Type "apropos word" to search for commands related to "word".
Attaching to process 625
Reading symbols from /usr/bin/python3.6...Reading symbols from /usr/lib
    /debug/.build-id/2c/3972a143bed2ede030627a64ce934ea4398f18.debug...
    done.
done.
[...]
(gdb)
```

```
(gdb) bt
#0  0x00007f0cbc6a66d6 in futex_abstimed_wait_cancelable (private=0,
    abstime=0x0,
    expected=0, futex_word=0x15b82e0) at ../sysdeps/unix/sysv/linux/
        futex-internal.h:205
#1  do_futex_wait (sem=sem@entry=0x15b82e0, abstime=0x0) at
    sem_waitcommon.c:111
#2  0x00007f0cbc6a67c8 in __new_sem_wait_slow (sem=sem@entry=0x15b82e0,
     abstime=0x0)
    at sem_waitcommon.c:181
#3  0x00007f0cbc6a6839 in __new_sem_wait (sem=sem@entry=0x15b82e0) at
    sem_wait.c:42
#4  0x000000000043f0a8 in PyThread_acquire_lock_timed (lock=lock@entry
    =0x15b82e0,
    microseconds=microseconds@entry=-1000000, intr_flag=intr_flag@entry
        =1)
    at ../Python/thread_pthread.h:354
#5  0x000000000058fbfd in acquire_timed (timeout=-1000000000, lock=0
    x15b82e0)
    at ../Modules/_threadmodule.c:68
```

```
#6  rlock_acquire (self=0x7f0cbc9c4660, args=<optimized out>, kwds=<
    optimized out>)
    at ../Modules/_threadmodule.c:314
#7  0x00000000004c549b in _PyCFunction_FastCallDict (kwargs=0x0, nargs
    =139692680693344,
    args=0x7f0cbb5ec870,
    func_obj=<built-in method acquire of _thread.RLock object at remote
         0x7f0cbc9c4660>)
    at ../Objects/methodobject.c:231
#8  _PyCFunction_FastCallKeywords (
    func=func@entry=<built-in method acquire of _thread.RLock object at
         remote 0x7f0cbc9c4660>, stack=stack@entry=0x7f0cbb5ec870, nargs
        =nargs@entry=0, kwnames=kwnames@entry=0x0)
    at ../Objects/methodobject.c:294
#9  0x000000000054ffe4 in call_function (pp_stack=pp_stack@entry=0
    x7ffeb109d4a8,
    oparg=<optimized out>, kwnames=kwnames@entry=0x0) at ../Python/
        ceval.c:4824
#10 0x00000000005546cf in _PyEval_EvalFrameDefault (f=<optimized out>,
    throwflag=<optimized out>) at ../Python/ceval.c:3322
```

Жемчужина

```
#11 0x000000000054f0e8 in PyEval_EvalFrameEx (throwflag=0,
    f=Frame 0x7f0cbb5ec6e8, for file /usr/lib/python3.6/logging/
        __init__.py, line 812, in acquire (self=<StreamHandler(filters
        =[], _name=None, level=0, formatter=<Formatter(_style=<
        PercentStyle(_fmt='%(levelname)s:%(name)s:%(message)s') at
        remote 0x7f0cbc962be0>, _fmt='%(levelname)s:%(name)s:%(message)s
        ', datefmt=None) at remote 0x7f0cbc962ba8>, lock=<_thread.RLock
        at remote 0x7f0cbc9c4660>, stream=<_io.TextIOWrapper at remote 0
        x7f0cbca7b708>) at remote 0x7f0cbc962b70>)) at ../Python/ceval.c
        :753
---Type <return> to continue, or q <return> to quit---
```

Используем Python extensions

```
(gdb) py-bt
Traceback (most recent call first):
  <built-in method acquire of _thread.RLock object at remote 0
      x7f0cbc9c4660>
  File "/usr/lib/python3.6/logging/__init__.py", line 812, in acquire
    self.lock.acquire()
  File "/usr/lib/python3.6/logging/__init__.py", line 861, in handle
    self.acquire()
  File "/usr/lib/python3.6/logging/__init__.py", line 1514, in
      callHandlers
    hdlr.handle(record)
  File "/usr/lib/python3.6/logging/__init__.py", line 1452, in handle
    self.callHandlers(record)
  File "/usr/lib/python3.6/logging/__init__.py", line 1442, in _log
    self.handle(record)
  File "/usr/lib/python3.6/logging/__init__.py", line 1306, in info
    self._log(INFO, msg, args, **kwargs)
  File "/usr/lib/python3.6/logging/__init__.py", line 1900, in info
    root.info(msg, *args, **kwargs)
```

Используем Python extensions (немного терпения)

```
1  File "/src/example.py", line 9, in foobar
2    logging.info('I am {}'.format(identity))
3  File "/usr/lib/python3.6/multiprocessing/process.py", line 93, in run
4    self._target(*self._args, **self._kwargs)
5  File "/usr/lib/python3.6/multiprocessing/process.py", line 258, in
       _bootstrap
6    self.run()
7  File "/usr/lib/python3.6/multiprocessing/popen_fork.py", line 73, in
       _launch
8    code = process_obj._bootstrap()
9  File "/usr/lib/python3.6/multiprocessing/popen_fork.py", line 19, in
       __init__
10   self._launch(process_obj)
11 File "/usr/lib/python3.6/multiprocessing/context.py", line 277, in
       _Popen
12   return Popen(process_obj)
13 File "/usr/lib/python3.6/multiprocessing/context.py", line 223, in
       _Popen
14   return _default_context.get_context().Process._Popen(process_obj)
```

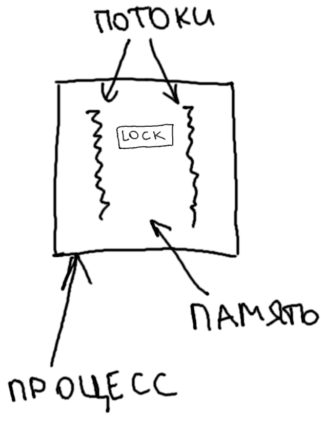
Используем Python extensions (вот оно)

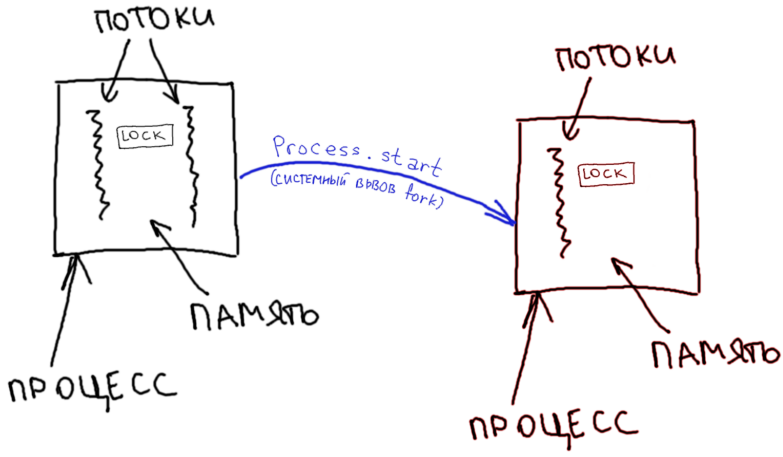
```
1  File "/usr/lib/python3.6/multiprocessing/process.py", line 105, in
       start
2    self._popen = self._Popen(self)
3  File "/src/example.py", line 20, in main
4    p.start()
5  File "/src/example.py", line 30, in <module>
6    main()
```

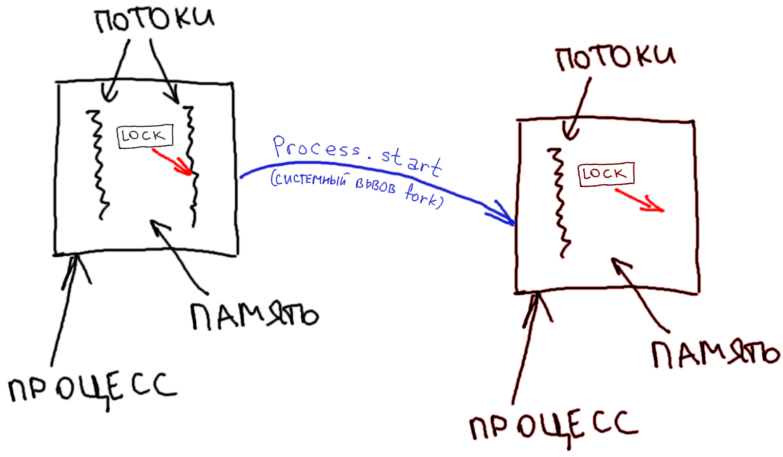
```
class Handler(Filterer):
    [...]
    def handle(self, record):
        """
        Conditionally emit the specified logging record.
        Emission depends on filters which may have been added to the handler.
        Wrap the actual emission of the record with acquisition/release of
        the I/O thread lock. Returns whether the filter passed the record for
        emission.
        """
        rv = self.filter(record)
        if rv:
            self.acquire()
            try:
                self.emit(record)
            finally:
                self.release()
        return rv
```

Листинг 1: logging/__init__.py#L851-L867

```python
import logging
from multiprocessing import Process
from threading import Thread
import time

logging.basicConfig(level='INFO')

def foobar(identity):
    for i in range(1000):
        logging.info('I am {}'.format(identity))
        time.sleep(0.001)

t = Thread(target=foobar, args=('thread',))
p = Process(target=foobar, args=('process',))

t.start()
p.start()

t.join()
logging.info('thread joined')
p.join()
logging.info('process joined')
```


ПОТОКИ
LOCK
ПАМЯТЬ
ПРОЦЕСС


ПОТОКИ
LOCK
ПАМЯТЬ
ПРОЦЕСС
Process.start
(системный вызов fork)
ПОТОКИ
LOCK
ПАМЯТЬ
ПРОЦЕСС


ПОТОКИ
LOCK
ПАМЯТЬ
ПРОЦЕСС
Process.start
(системный вызов fork)
ПОТОКИ
LOCK
ПАМЯТЬ
ПРОЦЕСС

Но как это касается лично меня?

Библиотека, которую вы используете,
может запускать потоки под капотом

- pymongo
- raven (sentry client)

Окей... но что делать?

1. Форкаться как можно раньше

```
process.start()
thread.start()
```

Окей... но что делать?

1. Форкаться как можно раньше

```
process.start()
thread.start()
```

2. Не использовать потоки в своем коде (заменить `threading` на `multiprocessing`)

Окей... но что делать?

1. Форкаться как можно раньше

```
process.start()
thread.start()
```

2. Не использовать потоки в своем коде (заменить `threading` на `multiprocessing`)
3. Разблокировать все lock'и после форка

```
def process_target(identity):
    logging._releaseLock()
    return foobar(identity)

process = Process(target=process_target, args=('process',))
process.start()
```

Что почитать?

- `man 2 fork`
- `https://wiki.python.org/moin/DebuggingWithGdb`
- `https://devguide.python.org/gdb/`

Стас Рудаков

mailto:stas@garage22.net

http://staaas.net/talks.html