

Нити и объекты синхронизации

Владимир Борисенко

Мехмат МГУ

vladimir_borisen@mail.ru

- Процесс — это задача, работающая на компьютере, которой выделяются следующие ресурсы:
 - Виртуальная память
 - Множество файловых дескрипторов
 - Множество нитей, работающих в рамках одного процесса
- Нить (thread), или легковесный процесс, или поток — это как бы подпроцесс внутри процесса. Нити выполняются параллельно или псевдопараллельно (в режиме разделения времени). Всем нитям доступны все ресурсы процесса, в рамках которого они выполняются. Нити разделяют
 - Общую виртуальную память. Адреса объектов одинаковы во всех нитях
 - Множество файловых дескрипторов процесса
 - Однако каждой нити выделяется **свой собственный стек**

Общие ресурсы нитей

- Разные нити выполняют совместную работу над общими данными. Обмен информацией между нитями возможен как через статические (глобальные) переменные, так и через данные в динамической памяти.
- Однако за счет того, что каждой нити выделяется свой стек (в общем адресном пространстве), **локальные переменные функций** свои в каждой нити. Это означает, что одна и та же функция может одновременно вызываться из разных нитей, и при этом они не мешают друг другу, поскольку разные нити работают с наборами локальных переменных в разных стеках. Иными словами, использование локальных переменных *безопасно при работе нитей* (thread safe).
- Напротив, использование **статических переменных** опасно, так как разные нити могут обращаться к ним в непредсказуемые моменты времени. Поэтому при работе с глобальными переменными для доступа к ним следует использовать **объекты синхронизации**

Объекты синхронизации

Популярны следующие объекты синхронизации, которые, как правило, поддерживаются на уровне ядра операционной системы:

- **Мьютексы** (mutex — MUTual EXclusive) используются для исключения одновременного доступа двух и более нитей к критическим данным. В Python'е вместо слова mutex используется термин Lock
- **Семафоры** исторически были первыми объектами синхронизации. Семафоры ограничивают доступ к общим данным заданным количеством нитей / процессов. Семафоры также могут использоваться для передачи информации о некотором событии от одной нити к другой.

- **События (event)** — частный случай семафоров, используются для сигнализации между нитями о том, что наступило некоторое событие.
- **Условные переменные (Condition)** как бы объединяют событие и мьютекс. Идея состоит в том, что нить, ожидающая событие, связанное с критическими данными, при получении сигнала о событии сразу получает и мьютекс, обеспечивающий ее исключительный доступ к критическим данным.

Нити в Python'е

В Python'е для работы с нитями используется модуль `threading`. Вот пример простейшей программы, запускающей две нити, осуществляющий жеребьевку. Первая нить 10 раз печатает слово `Heads`, вторая 10 раз печатает слово `Tails`. В промежутках между печатями нить ждет случайное время в интервале от 0.1 до 1 секунды. Мьютекс (`Lock` в Python'е) здесь используется для защиты печати, чтобы исключить одновременную печать двух нитей, что может привести к “каше” при выводе текста. Перед каждой печатью нить захватывает мьютекс и после печати освобождает его.

```
m.acquire()  
print(txt)  
m.release()
```

Простейшая программа, использующая нити

```
import threading
import time
import random

m = threading.Lock()

def draw(txt):
    for i in range(10):
        t = random.randrange(100, 1000)
        time.sleep(t/1000.)

        m.acquire()
        print(txt)
        m.release()
```

```
def main():
    t1 = threading.Thread(target=draw, args=("Heads",))
    t1.start()
    t2 = threading.Thread(target=draw, args=("Tails",))
    t2.start()
    t1.join()
    t2.join()

if __name__ == "__main__":
    main()
```