

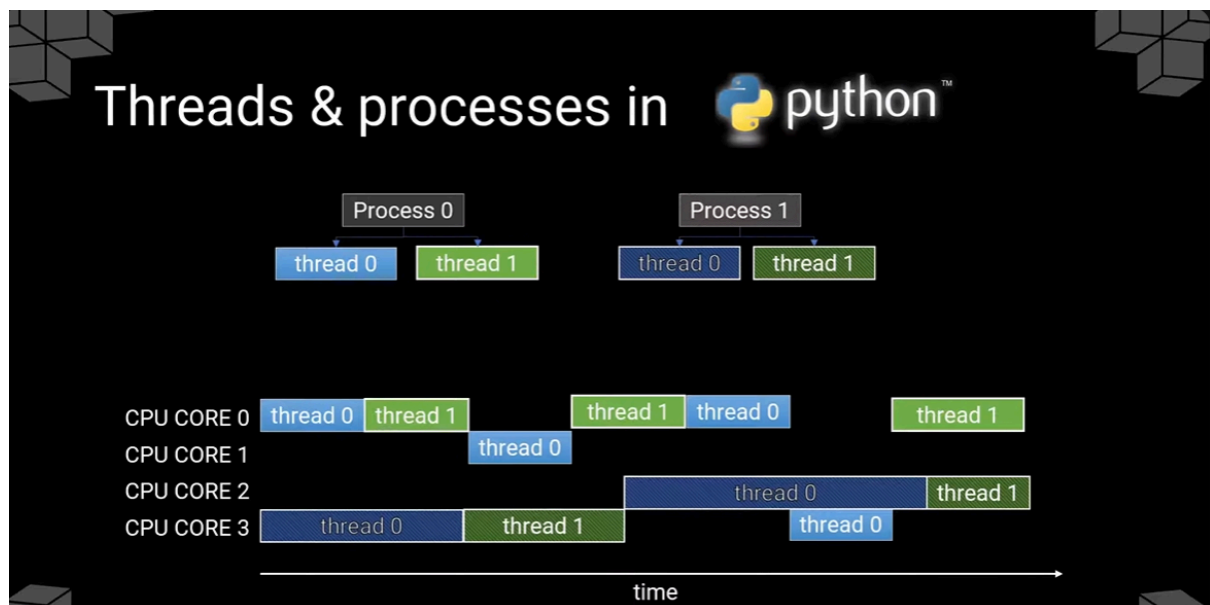
Python standart o'rnatuvchi **threading** kutubxonasini ishlatish orqali multithreading amalga oshiriladi.

threading kutubxonasidagi bazi asosiy obyektlar:

1. **Thread**: Bitta dasturni ko'p qatlamli qilish uchun ishlatiladi.
2. **Lock**: Qatlamning xavfsizligi uchun ishlatiladi. (Threadlar o'zaro resurslarni boshqarish uchun)
3. **Semaphore**: Concurrent jarayonlar sonini chegaralash uchun ishlatiladi.
4. **Event**: Bitta jarayonlarning ikki holati mavjudligini tekshirish uchun.
5. **Timer**: Belgilangan vaqtdan so'ng ishga tushiriladigan jarayonlar uchun.

Ko'p jarayonlar, shu jumlada, shu davrga qarab, parallel amallar, uzluksiz interaktiv tajriba va qo'llanuvchilar uchun oddiy vaqtli operatsiyalarni amalga oshirish uchun ishlatiladi.

Quyidagi misol bitta thread yaratish va unga ish bermoq uchun Python kodini ko'rsatadi:



THREADS VS MULTIPROCESSING

Time (ns)	Loop iters
0	0
50,000	0
100,000	0
150,000	0

```
def f(t0, array):
    dt = array[1, 0] - array[0, 0]
    while True:
        t = time.perf_counter_ns() - t0
        i = int(t/dt)
        if i >= len(array):
            break
        array[i, 1] += 1
    return None

import threading
t0 = time.perf_counter_ns()
for i in range(nthreads):
    arrays.append(np.copy(array))
    threads.append(threading.Thread(target=f, args=(t0, arrays[-1])))
```

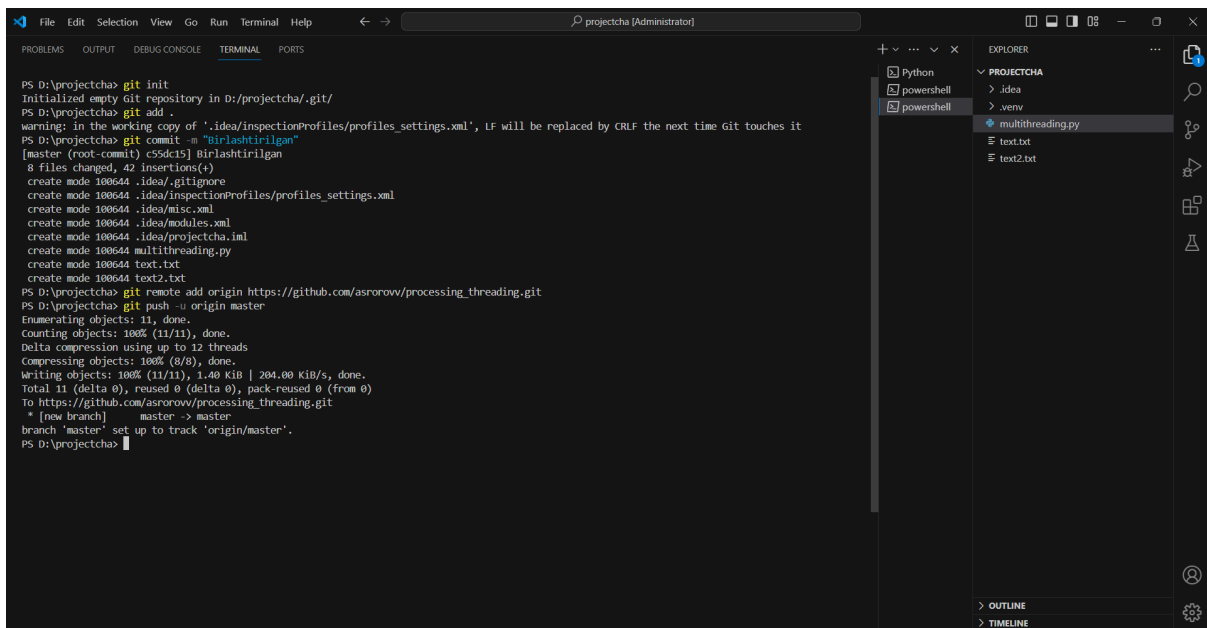
MULTIPROCESSING

Pythonning "Multiprocessing" moduli, bir nechta protsesslarni bitta dasturda eshitish va boshqarish imkonini beradi. Bu modul, Python dasturlarini to'g'ri kompyuterlar (kompyuterlar) uchun shunchaki o'rtacha. "Multiprocessing" moduli, bitta protsessdan foydalanish bilan solishtirishning unumdor vaqt iste'moli qiladi. Bu modulni ishlatish uchun foydalanuvchilarning kompyuterlarida ko'p yadroli protsessorni o'rtachasiz o'rnatish, chunki "Multiprocessing" modulida har bir protsess o'z resurslari bilan ishlashi uchun alohida yadro talab etadi.

"Multiprocessing" modulini ishlatish bilan, bir nechta protsesslarni bitta dasturda ishga tushirishingiz, ularning ishini bajarishini va natijalarni ko'rish imkoniyatiga egasiz.

Threads & processes in python™

	threading	multiprocessing
CPU limited tasks		✓
I/O bound tasks	✓	



```
PS D:\projectcha> git init
Initialized empty Git repository in D:/projectcha/.git/
PS D:\projectcha> git add
warning: in the working copy of '.idea/inspectionProfiles/profiles_settings.xml', LF will be replaced by CRLF the next time Git touches it
PS D:\projectcha> git commit -m "Birlashtirilgan"
[master (root-commit) c55dc15] Birlashtirilgan
 8 files changed, 42 insertions(+)
 create mode 100644 .idea/.gitignore
 create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
 create mode 100644 .idea/misc.xml
 create mode 100644 .idea/modules.xml
 create mode 100644 .idea/projectcha.iml
 create mode 100644 multithreading.py
 create mode 100644 text.txt
 create mode 100644 text2.txt
PS D:\projectcha> git remote add origin https://github.com/asrorovv/processing_threading.git
PS D:\projectcha> git push -u origin master
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (11/11), 1.40 KiB | 204.00 KiB/s, done.
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/asrorovv/processing_threading.git
 * [new branch]    master -> master
branch 'master' set up to track 'origin/master'.
PS D:\projectcha>
```

YUKLSAH JARAYONI

MALUMOTLAR INTERNETDAN OLINDI

