

# به نام خدا

## اعضای تیم

صبا کیانوش 400522157  
علی شیخ عطار 99542222

## استاد درس

دکتر سعید پارسا

## دستیار استاد

دکتر کاوان فرهادی بانه

این DSL در دامین برنامه نویسی قرار دارد. یکی از معضلات مهمی که برنامه نویس ها با آن دست و پنجه نرم می کنیم مولتی ترد پروگرامینگ یا برنامه نویسی parallel هست که به کمک این DSL با ورودی گرفتن برنامه ابتدا با استفاده از پروتکل هایی که هارد کد شده اند تشخیص میدهد که قابل اجرا با چند ترد میباشد یا اصلا صرفه ای دارد یا خیر در صورت تشخیص مثبت، نسخه ی ریوایز شده ی برنامه را با اجرای مالتی تردینگ بر روی آن خروجی میدهد

ورودی برنامه:

```
1 threads_no : n
2
3 time : TRUE/FALSE
4
5 Code : <python code address>
```

threads\_no : تعداد thread ای هست که قصد داریم برنامه ی ما توسط آن ها اجرا شود.

time : در صورت True بودن زمان اجرای برنامه هم در انتها پرینت میشود تا بتوانیم با زمانی که بدون پارالل کردن کد حاصل میشد مقایسه کنیم.

code : آدرس کد برنامه ی پایتون می باشد.

چند نمونه از عملکرد برنامه :

ورودی ۱:

```
def print_repeated(start, end, shared_sum):
    local_sum = 0
    for i in range(start, end):
        local_sum += i
        shared_sum[0] += local_sum

shared_sum = [0]

print_repeated(0,3)
print_repeated(3,7)
print(shared_sum[0])
print("Both threads have finished.")
```

خروجی ۱:

```
import threading

def print_repeated(start, end, shared_sum, lock):
    local_sum = 0
    for i in range(start, end):
        local_sum += i
        with lock:
            shared_sum[0] += local_sum

shared_sum = [0]
lock = threading.Lock()

thread1 = threading.Thread(target=print_repeated, args=(0, 3, shared_sum, lock))
thread2 = threading.Thread(target=print_repeated, args=(3, 7, shared_sum, lock))

thread1.start()
thread2.start()

thread1.join()
thread2.join()

print(shared_sum[0])
```

ورودی ۲:

```
def scrape_website(url, results):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')
    title = soup.find('title').get_text()
    results[url] = title

urls = [
    'https://example.com',
    'https://example.org',
    'https://example.net',
]

results = {}

for url in urls:
    scrape_website(url, results)

for url, title in results.items():
    print(f'{url}: {title}')
```

خروجی ۲:

```
def scrape_website(url, results, lock):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')
    title = soup.find('title').get_text()

    with lock:
        results[url] = title

urls = [
    'https://example.com',
    'https://example.org',
    'https://example.net',
    # Add more URLs as needed
]

results = {}
lock = threading.Lock()
threads = []

for url in urls:
    thread = threading.Thread(target=scrape_website, args=(url, results, lock))
    threads.append(thread)
    thread.start()

for thread in threads:
    thread.join()

for url, title in results.items():
    print(f'{url}: {title}')
```