

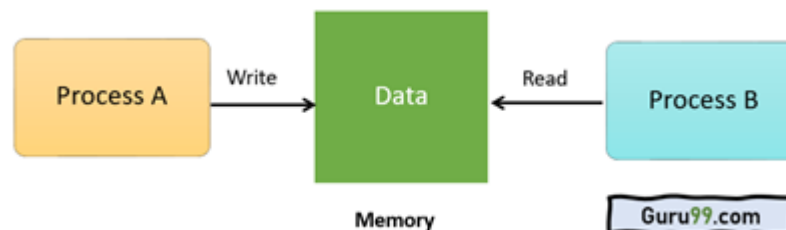
مینی پروژه سینکرونایز کردن فرآیندها

• طراح : حسام محبی ، فاطمه شفیعی

شرح پروژه

(Process Synchronization):

به عمل مدیریت اجرای چندین پراسس که به طور همزمان می خواهند به منابع مشترک مانند shared memory دسترسی داشته باشند گفته می شود .



به عنوان مثال دو پراسس همزمان قصد خواندن و نوشتن در یک داده مشترکی را دارند. که احتمال زیادی برای ایجاد خطا هنگام خواندن داده برای پراسس B وجود دارد .

برای جلوگیری از ایجاد خطا در چنین حالتی از mutex lock استفاده می شود . mutex lock اطمینان حاصل می کند که تنها یکی از دو پراسس در لحظه به داده مشترک دسترسی داشته باشند. وظیفه شما در این پروژه پیاده سازی یک lock ساده است که عمل مدیریت پراسس ها و جلوگیری از دسترسی همزمان به منبع مشترک را بر عهده داشته باشد.

در این پروژه چهار پراسس (add,sub,mul,div) تعریف شده است که هر یک دارای فرکانس خاص خود هستند. این فرکانس ها با یکدیگر تلاقی دارند. به عنوان مثال پراسس add از لحظه شروع در بازه های یک ثانیه ای به متغیر گلوبال دسترسی دارد و در این مدت پراسس دیگری نباید به متغیر گلوبال دسترسی داشته باشد. پس از پایان مدت یک ثانیه پراسس add متغیر گلوبال را آزاد کرده تا پراسس بعدی به آن دسترسی داشته باشد.

مدیریت این پراسس ها به نحوی است که برنامه نباید با پیام Process conflict مواجه شود. پیاده سازی این پروژه با زبان پایتون می باشد. شما در این پروژه می توانید از کلاس های زیر که در کتابخانه

multiprocessing پایتون موجود است استفاده کنید.

- **Process** : مستقل و موازی و شکل موازی و مستقل : Process . از برنامه اصلی اجرا می شود .
- **Value** : ساخت متغییر گلوبال را بر عهده دارد . شما در این پروژه محدودیتی در ایجاد هر : Value . تعداد متغییر گلوبال ندارید .
- **Array** : همانند یک متغییر گلوبال ساده قابلیت اشتراک بین چند پراسس را دارد و علاوه بر : Array را نیز دارد شما در این پروژه تنها محدود object مقادیر عددی قابلیت ذخیره شکل های دیگر مثل . به ایجاد یک آرایه گلوبال هستید .
- **Queue** : همانند یک متغییر گلوبال ساده قابلیت اشتراک بین چند پراسس را دارد و علاوه بر : Queue را نیز دارد شما در این پروژه تنها محدود object مقادیر عددی قابلیت ذخیره شکل های دیگر مثل . به ایجاد یک آرایه گلوبال هستید .

در قالب کد زنی، کدی که در اختیارتان قرار میگیرد، که چهار تابع به شکل مقابل که وظایف جمع،تفریق،ضرب و تقسیم رابرعده دارد.

• لینک فایل کد نمونه

شما در این پروژه آزاد هستید تا پارامتر های هرتابع را به هر شکلی که مناسب دیدید تغییر دهید.

```
1 |
2 | def add(num, value):
3 |     tmp = 0
4 |     while True:
5 |         print('add')
6 |         num.value += value
7 |         tmp = num.value
8 |         sleep(1)
9 |         if tmp != num.value:
10 |             print("Process conflict")
```

- تابع show() صرفا وظیفه نمایش مقدار متغییر گلوبال در هر سیکل را بر عهده دارد.

```
1 |
  | def Show(num):
```

```

2 | while True:
3 |     sleep(0.5)
4 |     print(num.value)
5 |
6 |

```

در بخش main فرآیند های و متغیر های گلوبال ساخته شده و شروع به کار برنامه از این بخش است.

```

1 | if __name__ == '__main__':
2 |     num = Value('d', 0.0)
3 |     arr = Array('i', range(2))
4 |     q = Queue()
5 |     p1 = Process(target=add, args=(num, 10))
6 |     p2 = Process(target=sub, args=(num, 5))
7 |     p3 = Process(target=mul, args=(num, 2))
8 |     p4 = Process(target=div, args=(num, 4))
9 |
10 |     show = Process(target=Show, args=(num,))
11 |     show.start()
12 |     sleep(1)
13 |     p1.start()
14 |     p2.start()
15 |     p3.start()
16 |     p4.start()

```

- در صورتی که خواستید با زبان دیگری پروژه را پیاده سازی کنید مشکلی ندارد اما دقت کنید به کلیت مسئله که استفاده از برنامه نویسی موازی و سینکرونایز کردن تردها یا پراسس ها آسیبی نرساند. (دقت کنید بعضی زبان ها اجازه استفاده از پراسس ها را نمی دهند یا کتابخانه های آن ها آپشن هایی که کتابخانه پراسس پایتون در اختیار شما قرار می دهد، را ندارند...)
- داکای کامل از توضیحات پروژه ارائه دهید و در آن تمامی مراحل انجام پروژه را توضیح دهید (شامل توابع، تقسیم کار و ...).
- بهتر است پروژه های خود را رو گیت هاب برده تا تقسیم کار شما مشخص باشد. (اندکی امتیاز نیز برای گیت در نظر گرفته خواهد شد!)