



مدرس: دکتر ابراهیمی مقدم
تاریخ تحویل: 14 دی

دانشکده‌ی مهندسی و علوم کامپیوتر
تمرین اول کامپیوتری
طراح سوالات: محمد بهرامی ، بردیا کریمی نیا

شبیه ساز تقاطع ترافیک:

در این تکلیف شما باید یک تقاطع 4 لاینه را پیاده سازی نمایید که در آن ماشین ها به 3 جهت در هنگام رسیدن به تقاطع حرکت می کنند (مستقیم ، چپ ، راست).
وظیفه شما پیاده سازی یک سیستم concurrency قوی برای عدم برخورد وسایل نقلیه می باشد.

سیستم شما باید 3 شرط زیر را تضمین نماید

1. **عملیات بدون تصادف:** وسایل نقلیه نباید در تقاطع با یکدیگر برخورد کنند.
2. **عدالت:** تمامی لاین‌ها و وسایل نقلیه باید به نوبت فرصت عبور داشته باشند.
3. **عملیات بدون بن‌بست (deadlock):** وابستگی‌های چرخه‌ای بین لاین‌ها و مسیرها نباید باعث ایجاد بن‌بست شوند
4. **بهره‌وری:** اگر مسیرهای وسایل نقلیه با یکدیگر تداخل نداشته باشند، چند وسیله نقلیه می‌توانند همزمان از تقاطع عبور کنند.

اهداف:

برای همگام سازی از روش round robin وزن دار برای اولویت‌دهی مناسب به لاین‌هایی که ترافیک سنگین‌تری دارند استفاده شود.

همچنین استفاده از mutex (قفل گذار) مناسب به گونه‌ای که حرکات همزمان وسایل نقلیه در مسیرهای بدون تداخل امکان‌پذیر باشد.

توضیحات لازم:

1. در هنگام ورود یک ماشین به تقاطع از مسیری 3 حرکت قابل انجام است:
 1. مستقیم به حرکت ادامه دهد
 2. به راست گردش کند
 3. به چپ گردش کند
2. هر لاین دارای یک میزان شدت ترافیک می باشد که هرچه شدت ترافیک بیشتر باشد باید سرویس دهی به آن لاین توسط سیستم شما بیشتر باشد. (یعنی باید بیشتر از بروز بن بست جلوگیری کند).
3. از شما انتظار می رود که به جای استفاده از یک قفل گذار (mutex) روی کل تقاطع هر 4 لاین تقاطع دارای mutex مخصوص خود باشند تا سیستم بهتر عمل کند.

کلاس های لازم:

کلاس vehicle:

```
import threading

class Vehicle(threading.Thread):
    def __init__(self, vehicle_id, vehicle_type, lane, route, crossing_time):
        threading.Thread.__init__(self)
        self.vehicle_id = vehicle_id
        self.vehicle_type = vehicle_type
        self.lane = lane
        self.route = route
        self.crossing_time = crossing_time

    def run(self):
        # do the logic of the concurrency here
        pass
```

منطق همگام سازی در بخش run پیاده سازی بشود.

توضیحات:

:vehicle_id.1

شناسه منحصر به فرد برای هر وسیله نقلیه

:vehicle_type.2

نوع وسیله نقلیه (normal , emergency) که وسایل نقلیه emergency اولویت بالاتری نسبت به وسایل نقلیه عادی دارند و اگر یک وسیله عادی و اضطراری بخواند به یک لاین بروند سیستم شما باید اول به وسیله با type emergency اجازه ورود بدهد.

:lane.3

شماره لاین وسیله نقلیه که از آن به تقاطع می آید.

:route.4

جهتی که وسیله نقلیه می خواهد در آن به محض ورود به تقاطع حرکت کند.(straight, left, right).

:crossing_time.5

زمان عبور از تقاطع.

توابع لازم:

1. منطق Weighted Round Robin

- تابع: `get_next_lane`
- توضیح: تعیین لاین بعدی برای سرویس دهی بر اساس منطق *round-robin* وزنی. وزن هر لاین از پیش تعریف شده است و باید به ترتیب مناسب چرخش کند.

2. تشخیص تداخل مسیر

- تابع: `can_enter_intersection`
- توضیح: بررسی کنید آیا مسیر وسیله نقلیه با سایر مسیرهای فعال در تقاطع تداخل دارد یا خیر. اگر تداخلی وجود ندارد، اجازه عبور بدهید.

3. قفل‌گذاری مسیر

- توابع: `enter_intersection` و `exit_intersection`
- توضیح: از قفل‌های مختص مسیر استفاده کنید تا تنها یک وسیله نقلیه بتواند همزمان یک مسیر خاص را اشغال کند. پس از خروج وسیله نقلیه، قفل را آزاد کنید.

4. همگام‌سازی برای عدالت

- تابع: `is_lane_turn`
- توضیح: اطمینان حاصل کنید که وسایل نقلیه بر اساس منطق round-robin و ترتیب رسیدنشان در صف پردازش می‌شوند.

محدودیت ها:

شما نمی‌توانید از چراغ‌های راهنمایی یا مکانیزم‌های کنترلی متمرکز استفاده کنید ، پیاده‌سازی شما باید امکان حرکت همزمان را در مسیرهای بدون تداخل فراهم کند.

موارد در هنگام تحویل:

- پیاده سازی ها حتما با زبان پایتون انجام شود.
- مدیریت مناسب بین وسایل نقلیه اضطراری و عادی

نحوه ارسال تمرین:

فایل ها را به صورت زیپ شده و به صورت زیر نامگذاری نمایید:

OS-CA1_[Student id]_[Student Name]

