بسمه تعالی

شبیه‌سازی کامپیوتری

پروژه درس شبیه‌سازی کامپیوتری

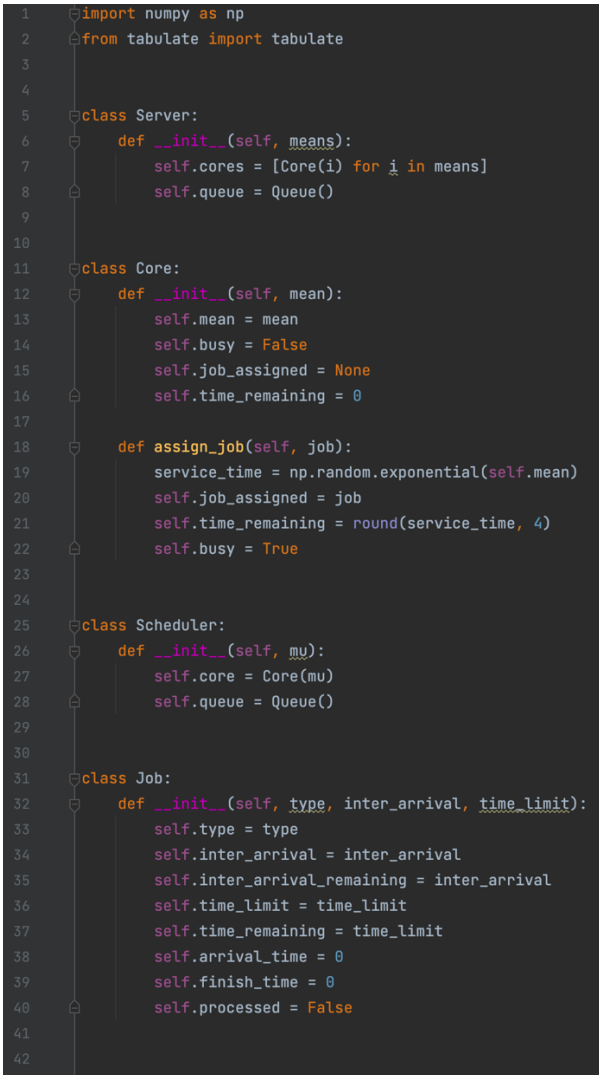
استاد

دکتر صفایی

ارشان دلیلی ۹۸۱۰۵۷۵۱

پروژه

ابتدا تصویری از کد را در زیر می‌بینید. داریم: (لازم به ذکر است که تمام کد در تصویر زیر نیامده است. هم‌چنین فایل کد پروژه به زبان پایتون در پوشه Project قرار دارد.)



```python
import numpy as np
from tabulate import tabulate


class Server:
    def __init__(self, means):
        self.cores = [Core(i) for i in means]
        self.queue = Queue()


class Core:
    def __init__(self, mean):
        self.mean = mean
        self.busy = False
        self.job_assigned = None
        self.time_remaining = 0

    def assign_job(self, job):
        service_time = np.random.exponential(self.mean)
        self.job_assigned = job
        self.time_remaining = round(service_time, 4)
        self.busy = True


class Scheduler:
    def __init__(self, mu):
        self.core = Core(mu)
        self.queue = Queue()


class Job:
    def __init__(self, type, inter_arrival, time_limit):
        self.type = type
        self.inter_arrival = inter_arrival
        self.inter_arrival_remaining = inter_arrival
        self.time_limit = time_limit
        self.time_remaining = time_limit
        self.arrival_time = 0
        self.finish_time = 0
        self.processed = False
```

حال هر قسمت از کد را در زیر بررسی می‌کنیم و سپ چند نمونه اجرای آن را برای ورودی‌های مختلف می‌آوریم. داریم:

ابتدا کلاس‌های موجود در کد را بررسی می‌کنیم. داریم: (لازم به ذکر است از دو کتاب‌خانه numpy و tabulate استفاده شده است. دلیل استفاده از کتاب‌خانه tabulate صرفاً مرتب‌کردن خروجی‌های خواسته شده است که در قالب جدول خروجی‌های خواسته شده برای حالت‌های مختلف داده شود و تأثیری در روند و منطق شبیه‌سازی ندارد.)

کلاس Server:

```
class Server:
    def __init__(self, means):
        self.cores = [Core(i) for i in means]
        self.queue = Queue()
```

این کلاس دارای هسته‌ها و هم‌چنین صف سرور است.

کلاس Core:

```
class Core:
    def __init__(self, mean):
        self.mean = mean
        self.busy = False
        self.job_assigned = None
        self.time_remaining = 0

    def assign_job(self, job):
        service_time = np.random.exponential(self.mean)
        self.job_assigned = job
        self.time_remaining = round(service_time, 4)
        self.busy = True
```

این کلاس دارای میانگین سرویس‌تایم هسته (میانگین توزیع نمایی)، وضعیت busy، کاری که در حال حاضر در حال خدمت‌رسانی به آن است و زمان باقی‌مانده تا اتمام آن کار است.

این کلاس هم‌چنین یک متد assign_job دارد که با دریافت یک کار، آن را به هسته تخصیص می‌دهد و هسته به انجام آن کار می‌پردازد.

(سرویس‌تایم از توزیع نمایی با میانگین هسته پیروی می‌کند و جهت خواناتر شدن اعداد تا ۴ رقم اعشار گرد شده‌اند.)

کلاس Scheduler:

```
class Scheduler:
    def __init__(self, mean):
        self.core = Core(mean)
        self.queue = Queue()
```

این کلاس دارای هسته و هم‌چنین صف زمان‌بند است.

کلاس Job:

```python
class Job:
    def __init__(self, type, inter_arrival, time_limit):
        self.type = type
        self.inter_arrival = inter_arrival
        self.inter_arrival_remaining = inter_arrival
        self.time_limit = time_limit
        self.time_remaining = time_limit
        self.arrival_time = 0
        self.finish_time = 0
        self.processed = False
```

این کلاس دارای نوع، زمان بین ورود، مهلت زمانی، زمان ورود، زمان اتمام و یک Boolean به نام processed برای پردازش شدن کار است. به طوری که اگر processed برابر True باشد، کار در هسته پردازش شده و به اتمام رسیده و از آن خارج شده است و در غیر این صورت برابر False است.