

به نام خدا



دانشکده‌ی مهندسی کامپیوتر

پروژه درس شبیه‌سازی کامپیوتری

نیم‌سال دوم ۱۴۰۱-۱۴۰۰

نکات مهم

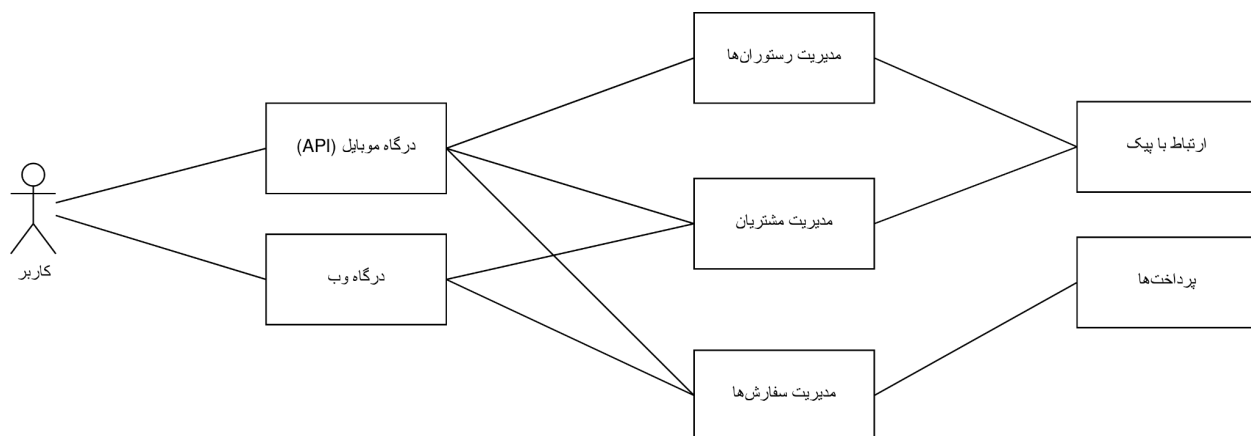
- موعد تحویل پروژه ساعت ۲۳:۵۹ روز ۳۱ تیر است.
- پروژه را در قالب تیم‌های حداکثر دو نفره انجام دهید.
- پروژه باید به صورت فایل zip با نام CS_Project_StudentNumber همان شماره دانشجویی است) در صفحه‌ی CW درس ارسال شود. همه‌ی فایل‌های مربوط به پاسخ باید در فایل زیپ فرستاده شوند.
- برای پرسیدن سوالات خود از طریق تالار گفت‌وگو در CW در ارتباط باشید. قبل از پرسش، سوالات جواب داده شده را مشاهده کنید؛ سوالات تکراری پاسخ داده نمی‌شوند.
- از نوشتن اطلاعات نامرتب در پاسخ‌های خود پرهیز کنید.
- همفکری بین دانشجویان درس برای حل تمرین ایرادی ندارد، اما هر کس باید پاسخ خودش را بنویسد. در صورت مشاهده‌ی موارد مشکوک به تقلب، مطابق با قوانین کمیته آموزش دانشکده برخورد خواهد شد.

پروژه پایانی

در این پروژه شما باید یک سیستم مبتنی بر میکروسرویس را به صورت رخداد گسسته شبیه‌سازی کنید و خروجی‌های مورد نظر برای ارزیابی سیستم را بدست آورید.

توضیح سیستم

یک سیستم درخواست غذای آنلاین مثل اسنپ‌فود را در نظر بگیرید. این سیستم از تعدادی میکروسرویس تشکیل شده‌است که هر کدام وظیفه مشخصی دارند و با هماهنگی یکدیگر به درخواست‌های کاربران جواب می‌دهند. در ادامه یک معماری فرضی را برای این سیستم مشاهده می‌کنید:



یک کاربر می‌تواند از طریق کلاینت موبایل و یا وب با سیستم ارتباط برقرار کند. هر سرویس¹ با سرعت مشخصی درخواست‌ها را پردازش کرده و جواب می‌دهد. همچنین هر نوع سرویس یک صف دارد که با نظم مشخصی درخواست‌ها را دریافت می‌کند. از هر سرویس ممکن است یک یا بیشتر نمونه² وجود داشته باشد که درخواست‌ها به صورت تصادفی به یک نمونه‌ی بیکار فرستاده می‌شود.

مثلا سرویس "ارتباط با پیک" را در نظر بگیرید و فرض کنید ۴ نمونه‌ی در حال اجرا از این سرویس وجود دارد. این ۴ سرویس یک صف برای پردازش درخواست‌ها دارند. حال به ازای هر درخواست جدید، اگر تعدادی از این ۴ نمونه بیکار باشند، به صورت تصادفی این درخواست به یکی از آن‌ها تخصیص داده می‌شود، در غیر این صورت این

¹ مستطیل‌ها نشان‌دهنده میکروسرویس‌ها هستند.

² instance

درخواست وارد صفِ مربوط به سرویس "ارتباط با پیک" شده و منتظر می‌ماند تا یکی از این ۴ تا کارش به اتمام رسیده و پردازش درخواست را شروع کند.

هر نمونه از یک سرویس، پس از اتمام کار خود، یک درخواست را (در صورت وجود) از صف مربوطه برداشته و پردازش می‌کند. اطلاعات دقیق‌تر سرویس‌ها از جمله ترتیب پردازش صف‌ها در جدول زیر آمده است.

وابستگی‌هایی بین این سرویس‌ها وجود دارد. مثلاً سرویس "مدیریت رستوران‌ها" برای اتمام درخواست خود، درخواستی را ممکن است به سرویس "ارتباط با پیک" ارسال کند و منتظر جواب آن بماند. اگر سرویس A برای جواب به درخواست خود، وابسته به سرویس B است، ابتدا درخواستی را به سرویس B ارسال می‌کند، سرویس B با توجه به زمان سرویس‌دهی خود، مدتی بعد جواب را به A ارسال می‌کند، حال سرویس A شروع به ادامه کار می‌کند و بعد از زمان سرویس‌دهی خود، درخواست را به اتمام می‌رساند (پس به طور کلی زمان رسیدگی به این درخواست در سیستم، حاصل جمع مدت زمان سرویس‌دهی A و B است).

در این جدول مشخصات سرویس‌ها آورده شده است:

جدول ۱

نام سرویس	وظیفه	مدل صف	میانگین زمان سرویس‌دهی ^۳ (ثانیه) ^۴
مدیریت رستوران‌ها	مدیریت اطلاعات و عملیات مربوط به رستوران و رستوران‌داران	FIFO	۸
مدیریت مشتریان	مدیریت اطلاعات و عملیات مربوط به مشتریان	FIFO	۵
مدیریت سفارش‌ها	مدیریت ثبت و پیگیری سفارش	FIFO	۶
ارتباط با پیک	درخواست پیک و ارتباط با او	FIFO	۹
پرداخت‌ها	انجام عملیات مربوط به پرداخت	FIFO	۱۲
درگاه API	ارائه سرویس‌ها به کلاینت موبایل	FIFO	۲
درگاه وب	ارائه سرویس‌ها به کلاینت وب	FIFO	۳

^۳ مدت زمان سرویس‌دهی از توزیع نمایی پیروی می‌کند. عدد بیان‌شده، میانگین این توزیع را مشخص می‌کند. همچنین دقت کنید که این عدد، بیانگر زمان سرویس بدون در نظر گرفتن وابستگی‌های آن به سایر سرویس‌ها است.

^۴ برای سهولت در پیاده‌سازی، از اعداد بزرگ استفاده شده است. در واقعیت این اعداد از مرتبه‌ی میلی ثانیه هستند.

در این جدول مشخصات درخواست‌ها از سمت کاربر آورده شده است:

جدول ۲

عنوان درخواست	زنجیره‌ی وابستگی‌های بین سرویسی	احتمال وقوع (درصد)
ثبت سفارش با موبایل	درگاه موبایل <-- مدیریت سفارش‌ها <-- پرداخت‌ها	۲۰
ثبت سفارش با وب	درگاه وب <-- مدیریت سفارش‌ها <-- پرداخت‌ها	۱۰
ارسال پیام به پیک	درگاه موبایل <-- مدیریت مشتریان <-- ارتباط با پیک	۵
مشاهده اطلاعات رستوران از طریق موبایل	درگاه موبایل <-- مدیریت رستوران‌ها	۲۵
مشاهده اطلاعات رستوران از طریق وب	درگاه وب <-- مدیریت رستوران‌ها	۱۵
درخواست پیک	درگاه وب <-- مدیریت رستوران‌ها <-- ارتباط با پیک	۲۰
پیگیری سفارش	درگاه موبایل <-- مدیریت سفارش‌ها	۵

- نرخ کل درخواست‌هایی که به سیستم وارد می‌شوند به عنوان پارامتر ورودی به شما داده می‌شود. واحد آن تعداد درخواست در ثانیه است.

موارد امتیازی

نرخ خطا

برای هر سرویس، یک نرخ خطا⁵ در نظر می‌گیریم. به این صورت که هر درخواست در حال پردازش در سرویس A، پس از هر ثانیه⁶ که سپری می‌شود، با احتمال مشخصی، دچار خطا می‌شود. وقتی که درخواست دچار خطا شد، کار آن در سرویس A در همان لحظه به پایان می‌رسد و جواب دارای خطا به صدا کننده سرویس، برگردانده می‌شود (اگر سرویس A یک درخواست به سرویس B بفرستد و این درخواست در B دچار خطا شود، درخواست منتظر در A نیز دچار خطا می‌شود).

جدول نرخ خطای سرویس‌ها:

در این جدول مشخصات سرویس‌ها آورده شده است:

نام سرویس	نرخ خطا ⁷
مدیریت رستوران‌ها	۰.۰۲
مدیریت مشتریان	۰.۰۲
مدیریت سفارش‌ها	۰.۰۳
ارتباط با پیک	۰.۱
پرداخت‌ها	۰.۲
درگاه API	۰.۰۱
درگاه وب	۰.۰۱

⁵ error rate

⁶ واحد زمانی اصلی در این پروژه ثانیه است.

⁷ اگر برابر با p باشد، در هر ثانیه با احتمال p، درخواست در حال پردازش در این سرویس، دچار خطا می‌شود.

اولویت درخواست‌ها

برای درخواست‌ها اولویت در نظر بگیرید. این اولویت‌ها هنگام انتخاب درخواست از صف توسط یک سرویس در نظر گرفته می‌شوند. درخواست‌ها دارای دو سطح اولویت هستند، سطح ۱ اولویت بالاتری نسبت به سطح ۲ دارد.

عنوان درخواست	سطح اولویت
ثبت سفارش با موبایل	۱
ثبت سفارش با وب	۱
ارسال پیام به پیک	۲
مشاهده اطلاعات رستوران از طریق موبایل	۲
مشاهده اطلاعات رستوران از طریق وب	۲
درخواست پیک	۱
پیگیری سفارش	۲

حداکثر زمان انتظار درخواست‌ها

هر درخواست، حداکثر مدت زمانی برای انتظار خواهد داشت و اگر کاربر بیش‌تر از آن زمان منتظر بماند، درخواست را ناموفق در نظر می‌گیریم.⁸ این زمان‌ها برای هر درخواست به شکل پارامتر به شما داده می‌شوند.

⁸ Timeout

ورودی‌ها

این پارامترها به عنوان ورودی به شما داده می‌شوند⁹:

- تعداد نمونه‌ها از هر نوع سرویس
- نرخ ورود درخواست‌ها به سیستم (تعداد درخواست در ثانیه)
- مدت زمان شبیه‌سازی (به ثانیه)
- حداکثر زمان انتظار برای هر نوع درخواست (امتیازی)

نمونه ورودی¹⁰

دقت کنید که در سطر اول ترتیب نوع سرویس‌ها مانند جدول ۱ است. (یعنی عدد اول مربوط به سرویس مدیریت رستوران‌ها، عدد دوم مربوط به سرویس مدیریت مشتریان و ... است.) در سطر چهارم نیز ترتیب نوع درخواست‌ها مانند جدول ۲ است. (یعنی عدد اول مربوط به درخواست ثبت سفارش با موبایل، عدد دوم مربوط به درخواست ثبت سفارش با وب و ... است.)

1 1 1 2 5 3 2

30

28800

25 30 25 30 30 40 20

خروجی‌ها

یک توضیح مختصر از کد زده شده به همراه نحوه اجرای آن تهیه کنید. همچنین بعد از شبیه‌سازی این سیستم، به ازای هر دسته از ورودی‌های داده شده، این خروجی‌ها را در گزارش خود بیاورید:

- میانگین طول صف‌ها
- میانگین زمان صرف شده در صف‌ها (به طور کلی و به ازای هر درخواست)
- میزان بهره‌وری سرویس‌ها¹¹ (چه کسری از زمان مشغول بوده‌اند؟)
- تاثیر تغییر در تعداد نمونه‌های سرویس‌ها در زمان درخواست‌ها، طول صف‌ها و...
- پیشنهاد خود برای بهبود معماری گفته شده را بنویسید (کدام سرویس bottleneck بوده است؟).
- درصد درخواست‌های مواجه شده با خطا به تفکیک درخواست و کلی (امتیازی)
- درصد درخواست‌های منقضی شده¹² به تفکیک درخواست‌ها و کلی (امتیازی)

⁹ نیازی به دریافت ورودی‌ها از کنسول در برنامه نیست، می‌توانید آن‌ها را به شکل متغیر در کد تعریف کنید.

¹⁰ البته می‌توانید این ورودی را نیز تغییر دهید ولی مهم این است که با دریافت این ورودی‌ها سیستم شما کار را انجام دهد.

¹¹ Utilization

¹² Timeout

نکات پیاده سازی

- برای پیاده سازی استفاده از هر زبانی ممکن است.
- بهتر است ابتدا ساختار شی گرای مناسبی برای موجودیت های سیستم در نظر گرفته شود.
- در پایتون می توانید از کتابخانه های Ciw، SimPy، و ... استفاده کنید.
- استفاده از کتابخانه های دیگر، به گونه ای که شبیه سازی را ساده تر کند، توصیه می شود.
- برای راحتی در پیاده سازی، زمان ها را گسسته سازی کنید، یعنی اگر مدت زمان یک سرویس به شکل عددی دارای اعشار محاسبه شد، آن را گرد کنید. همچنین همه ی واحدهای زمانی را ثانیه در نظر بگیرید.
- بعنوان چند مثال و آشنایی بیشتر با شبیه سازی در پایتون می توانید این لینک ها را مشاهده کنید (مسئله های مطرح شده در این منابع با مسئله ی ذکر شده تفاوت دارد، هدف اصلی آشنایی با معماری کد و نحوه پیاده سازی ها است)
- <https://towardsdatascience.com/simulating-a-queueing-system-in-python-8a7d1151d485>
- <https://towardsdatascience.com/introduction-to-simulation-with-simpy-322606d4ba0c>
- <https://pythonhosted.org/SimPy/Tutorials/TheBank.html>
- <https://medium.com/analytics-vidhya/simulating-a-single-server-queueing-system-in-python-f8e32578749f>
- برای مثال، چند لینک زیر برای آشنایی با کتابخانه ی SimPy در پایتون به شما معرفی شده است و می توانید از آنها استفاده نمایید.
- <https://realpython.com/simpy-simulating-with-python/>
- <https://www.youtube.com/watch?v=NypbxgylScM>
- <https://simpy.readthedocs.io/en/latest/examples/>
- برای آشنایی با کتابخانه ی ciw نیز میتوانید به لینک زیر مراجعه کنید.
- <https://ciw.readthedocs.io/en/latest/>