
بسم الله الرحمن الرحيم

گزارش تمرین اول ارزیابی کارایی سیستم‌های کامپیوتری

گزارش

مبینا کاشانیان

۱۴۰۰-۱۴۰۱

فهرست مطالب

۱ گزارش تمرین شبیه سازی

۱.۱ چکیده

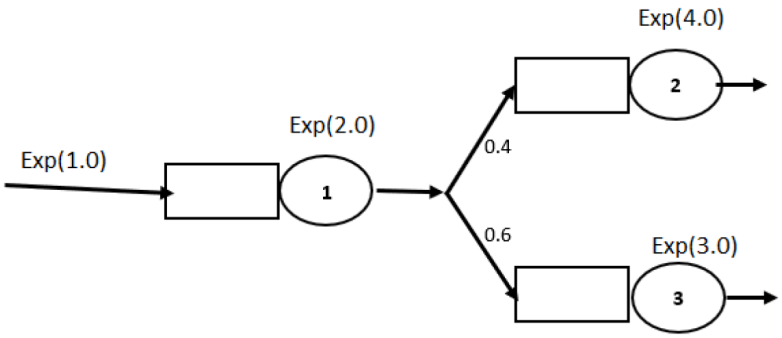
در پروژه اول درس کارایی سیستم‌های کامپیوتری باید با استفاده از زبان برنامه نویسی دلخواه یک مکانیزم صف را پیاده سازی میکردیم و معیارهای کارایی را برای هر صف محاسبه میکردیم. همچنین در این پروژه پیاده سازی UI امتیازی محسوب میگردید که در انتهای این مستند عکس های رابط کاربری را مشاهده میکنید. شکل ۶.۱

۲.۱ شرح دقیق مسئله

صفی که باید آن را شبیه سازی میکردیم در شکل ۱.۱ قابل مشاهده است. این شبکه شامل ۳ صف هست که با نرخ متفاوت و تابع نمایی کار میکند

۳.۱ پیاده سازی

برای شروع کار اگر فرض کنیم که شبکه دارای یک صف است پیاده سازی و تحلیل مسئله آسان تر خواهد شد.بنابراین در ابتدا پارامترهای ورودی شامل زمان های ورود و زمان های سرویس مقداردهی اولیه شده و بصورت لیست به یک تابع به نام System



شکل ۱.۱: صورت مسئله

```java
public double generateExponential(double lambda) {
    if (lambda == 0.0)
        return 0.0;
    double randomize;
    randomize = random.nextDouble();
    return -1/ (lambda)*Math.log(randomize) ;
}
```

شکل ۲.۱: تابع توزیع نمایی

states داده می شود که این تابع تمامی مقدارهای لازم شامل initializeSystemParameters اجرا run() را صفر کرده و سیستم را آماده پذیرش تسک میکند. سپس تابع Statical، counters و arrivalflag() بر اساس نوع رخداد تابع های مربوطه شامل clock می شود که در هر صدا زده شده و وضعیت سیستم به روز می شود که در هر مرحله پارامترهای exitedflag() لازم برای ارزیابی کارایی مقداردهی شده و در انتها با انجام محاسبات تمامی معیارهای کارایی بدست می آید. برای اینکه از عملکرد کد اطمینان حاصل کنم در ابتدای تشکیل صف، پارامترهای ورودی با مثالی که در اسلایدهای درس موجود است مقداردهی شدند و تمامی معیارهای کارایی به درستی ارزیابی و نمایش داده شده اند سپس به سراغ پیاده سازی تابع نمایی و تابع تولید عدد رندوم و تولید احتمال رفتم و آنها را پیاده سازی کردم و نرخ ورودی تابع نمایی را متناسب با اعداد داده شده در متن پروژه انتخاب کردم. پس از اینکه توابع مورد نیاز برای پیاده سازی را پیاده سازی کردم به سراغ متغیر های مورد نیاز در پروژه رفتم و آنها را برای هر یک از صف ها تخصیص دادم که این متغیر ها در واقع همان معیار های کارایی سیستم مورد نظر هستند و با توجه به اینکه ۳ صف در اختیار داریم ۳ کلاس جداگانه برای هر یک ساختم.

۱.۳.۱ پیاده سازی تابع توزیع نمایی

برای پیاده سازی تابع نمایی از الگوریتم زیر استفاده کردم و نرخ لامبدا را با توجه به صورت سوال مشخص کردم. که در شکل ۲.۱ قابل مشاهده است. برای هر تسک یک احتمالی در نظر گرفتم و اینکار را برای این اینجام دادم که بتوانم پس از خروج از صف اول تسک ها را به صف دوم یا صف سوم هدایت کنم . اگر نرخ احتمال تعیین شده برای صف بزرگتر از ۶.۰ باشد به صف سوم میرود و اگر از ۶.۰ کوچکتر باشد یعنی از ۴.۰ کوچکتر باشد در صف دوم جای دارد. سپس معیار های کارایی برای هریک را محاسبه کردم و در متغیر های کلاس هر صف ذخیره کردم.

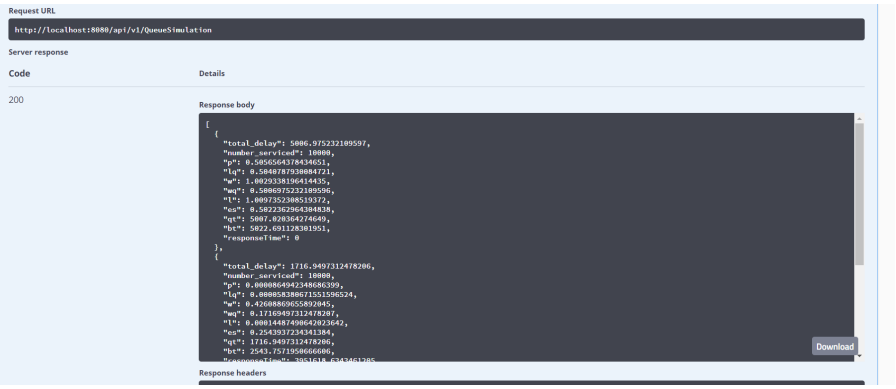
۲.۳.۱ نحوه اجرای کد

ابتدا mobinaapplication را اجرا کنید و سپس با استفاده از مرورگر خود وارد آدرس

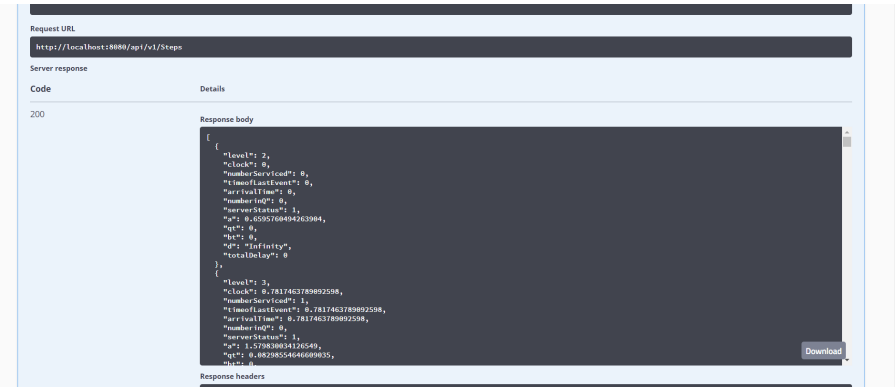
$$http://localhost:8080/swagger-ui.html$$

شوید و سپس بخش کنترلر را انتخاب کنید. در قدم بعدی ابتدا

$$@GetMapping("/api/v1/QueueSimulation")$$



شکل ۳.۱: خروجی هر صف



شکل ۴.۱: قدم به قدم محاسبه پارامتر ها

را اجرا کنید انچه که در خروجی سرور مشاهده میکنید به ترتیب پارامتر های کارایی هر صف میباشد. پس از آن

$@GetMapping("/api/v1/Steps")$

را اجرا کرده آنچه که در خروجی مشاهده میکنید مرحله به مرحله ی محاسباتی است که در هر صف انجام شده است.

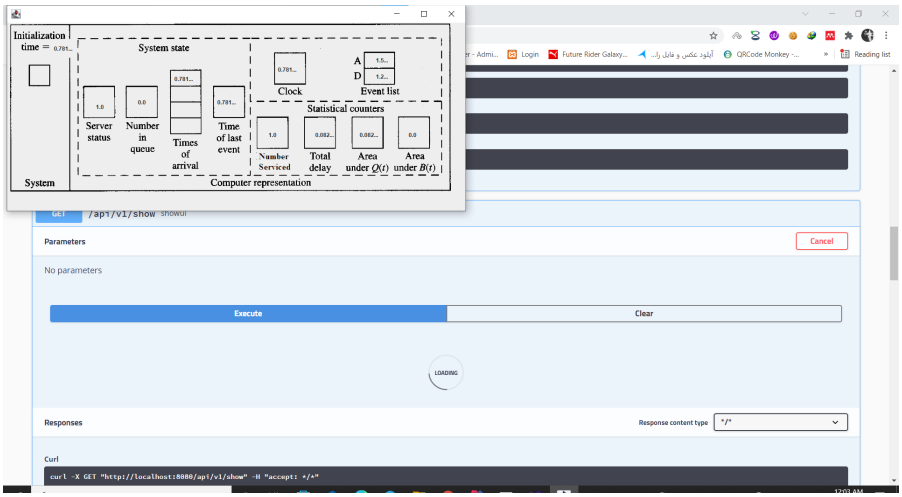
و در نهایت برای اجرای رابط کاربری

$@GetMapping("/api/v1/show")$

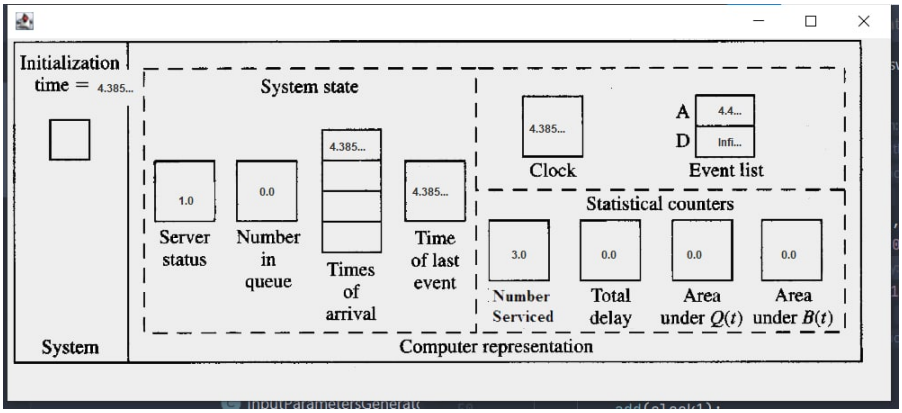
را انتخاب کنید و آنچه که در خروجی میبینید قدم به قدم محاسبه ی عملکرد کد برای به دست آوردن معیار کارایی میباشد.

۳.۳.۱ رابط کاربری UI (بخش امتیازی)

برای این پروژه من از زبان جاوا استفاده کردم و همچنین با استفاده از boot spring و swing برای این پروژه رابط کاربری پیاده سازی کردم که مرحله مرحله خروجی هر یک از صف ها را نشان می دهد. خروجی که در شکل ۶.۱ مشاهده میکنید رابط کاربری است که برای پروژه طراحی کرده ام.



شکل ۵.۱: خروجی کد به همراه رابط کاربری



شکل ۶.۱: رابط کاربری پروژه