



درس ارزیابی کارایی سیستم‌های کامپیوتری

دکتر علی موقر

تمرین ۲: شبیه‌سازی صف

**M M 1 K**

۴۰۰۲۱۱۲۱۳-محمد صابری<sup>۱</sup>

<sup>۱</sup> دانشجوی کارشناسی ارشد، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف

۲۷ آذر ۱۴۰۱

## ۱. تمرین اول

در این تمرین از ما خواسته شده که یک صف با چهارده خانه پیاده‌سازی کرده و عملکرد ورود و خروج و سرویس‌دهی به مشتریان را شبیه‌سازی کنیم. هر مشتری دارای ویژگی‌هایی مانند زمان ورود به صف، زمان حیات و مدت زمان مورد نیاز برای سرویس دارد. اگر مشتری وارد صف شود از لحظه ورودش به صف تا پایان مدت زمان حیاتش سرویس دریافت نکند از صف اخراج یا همان *Drop* می‌شود، اگر صف پر باشد و مشتری زمان ورودش به صف برسد باعث بلاک شدن مشتری می‌شود. حال ما در تمرین نرخ ورود، نرخ سرویس‌دهی، مدت زمان سرویس و ... داریم که در جدول زیر بنده پارامترهایی که در کد استفاده کردم را معرفی و بعد به توضیح کد می‌پردازیم.

Parameter in Assignment	Parameter in Code
$\mu$	Service_Rate
$\lambda$	Entry_Rate
$\theta$	Live_Time
Customer Processing Time Required	Service_Time
Time of Customer Enters the Queue	Arrival_Time
Arrival_Time + Live_Time	Eviction_Time

در ابتدا ما یک آرایه به نام *Queue* به عنوان صف مورد نظر با ۱۴ خانه تعریف کرده‌ایم، پس از آن در یک حلقه که از  $\lambda = 0.1$  شروع شده و تا  $\lambda = 20$  ادامه دارد کار شبیه‌سازی را آغاز می‌کنیم. در ابتدا به ازای هر لامبدا یا همان نرخ ورود مشتری به سیستم ما مشتریان را ایجاد و مقادیر

Arrival\_Time , Live\_Time , Service\_Time , Eviction\_Time ( Arrival\_Time + Live\_Time)

را برای هر مشتری محاسبه و برای آن‌ها اعطا می‌کنیم، سپس همه مشتری‌ها را آرایه *Customers* قرار می‌دهیم. برای آنکه بتوان عملکرد صف را به صورت رخدادی شبیه‌سازی کرد *Arrival Time* و اندیس آن مشتری در آرایه *Customers* را داخل یک *Min Heap* ذخیره می‌کنیم تا رخدادهای ورود را به صورت صعودی از کوچک به بزرگ بخوانیم و نسبت به هر رخداد ورود مشتری به صف کارهای مورد نیاز مانند *Drop* , *Block* , *Service* مشتری‌ها را انجام دهیم، اما برای آنکه بتوان در

شبیه‌سازی به ویژگی‌های هر مشتری دسترسی داشته باشیم اندیس آن در آرایه را در *Min Heap* قراره داده تا به ویژگی‌های مشتری مورد نظر دسترسی داشته باشیم.

حال به تعداد اندازه *Min Heap + Queue* یک حلقه را تکرار کرده تا تکلیف تمامی مشتری‌ها را مشخص نماییم. در ابتدا دو پارامتر مهم داریم، یکی از آن‌ها *Free Time* می‌باشد در واقع این پارامتر نشان می‌دهد که سرویس‌دهنده در چه زمانی بیکار می‌شود و می‌تواند به مشتری بعدی سرویس دهد و پارامتر بعدی ما *Time* است که در واقع زمان آن ما را در شبیه‌سازی مشخص می‌کند. در ابتدا با بررسی شود که مشتری‌های در صف زمان اخراجشان رسیده یا خیر، اگر رسیده باشد مشتری اخراج شده در غیر این صورت داخل صف می‌ماند. در هنگام وارد شدن مشتری به صف ما پارامتر *Time* را برابر زمان ورود آن مشتری به صف *Arrival Time* قرار می‌دهیم. حال پس از آنکه *Time* مشخص شده به بررسی وضعیت اخراج مشتری‌ها می‌پردازیم. اگر بعد از اخراج مشتری‌ها صف خالی بود مشتری وارد صف شده در غیر این صورت مشتری بلاک می‌شود. حال وقتی مشتری در صف وارد شد باید بررسی شود که مشتری قبلی که در حال سرویس گرفتن از سرویس‌دهنده بوده کارش تمام شده یا خیر. اگر *Free Time* ما کوچکتر و یا مساوی با *Time* ما باشد یعنی کار آن مشتری تمام شده و باید مشتری بعدی بر اساس قانون *FCFS* سرویس بگیرد. حال برای آنکه سرویس را به مشتری بعدی بدهیم *Free Time* را به صورت  $Free Time = Free Time + Customer Service Time$  محاسبه می‌کنیم تا لحظه بیکاری بعدی سرویس‌دهنده را بدست آوریم. برای آنکه یک مشتری را از صف اخراج کنیم باید شرط  $Customer Eviction Time \leq Time$  بررسی کنیم که اگر زمان اخراج مشتری کمتر یا مساوی زمان حال شبیه‌سازی بود از صف اخراج شود.

همچنین برای محاسبه و بدست آوردن میانگین تعداد مشتری‌های داخل صف، با هر رخداد یا همان رسیدن زمان ورود یک مشتری، در هر بخش یعنی تمام شدن سرویس هر مشتری یا بلوک شدن هر مشتری و یا اخراج هر مشتری از صف تعداد مشتری‌های داخل صف را شماره و در متغیری خیره می‌کنیم و در انتها متغیر را بر تعداد مشتری‌های کل در آن لامبدا تقسیم کرده تا  $N_e$  مورد نظر بدست آید.

Condition of Eviction( Drop ) Customers from Queue  $\Rightarrow$  Customer Eviction Time  $\leq$  Time

Condition of Exit Customer from Queue  $\Rightarrow$  Free Time  $\leq$  Time

Time = Arrival Time of New Customer

Free Time = Free Time + Service Time of Customer That in Cell 0 of Queue

در فایلی که تحویل داده شده برای دو حالت نمایی و ثابت زمان انتظار مشتری (*Live Time*) قرار دارد. خروجی‌های ما برای

حالت ثابت در شش فایل و حالت نمایی نیز در شش فایل ذخیره می شود.

### Exponential

$P_d$  (Probability of Drop in Simulation)  $\Rightarrow$  Drops-Simulate(Exp).txt

$P_d$  (Probability of Drop in Analytic)  $\Rightarrow$  Drops(Exp).txt

$P_b$  (Probability of Blocks in Simulation)  $\Rightarrow$  Blocks-Simulate(Exp).txt

$P_b$  (Probability of Blocks in Analytic)  $\Rightarrow$  Blocks(Exp).txt

$N_c$  (The Average Number of Customers in Queue in Simulation)  $\Rightarrow$

Customers\_In\_Queue-Simulate(Exp).txt

$N_c$  (The Average Number of Customers in Queue in Analytic)  $\Rightarrow$  Customers\_In\_Queue(Exp).txt

### Fixed

$P_d$  (Probability of Drop in Simulation)  $\Rightarrow$  Drops-Simulate(Fixed).txt

$P_d$  (Probability of Drop in Analytic)  $\Rightarrow$  Drops(Fixed).txt

$P_b$  (Probability of Blocks in Simulation)  $\Rightarrow$  Blocks-Simulate(Fixed).txt

$P_b$  (Probability of Blocks in Analytic)  $\Rightarrow$  Blocks(Fixed).txt

$N_c$  (The Average Number of Customers in Queue in Simulation)  $\Rightarrow$

Customers\_In\_Queue-Simulate(Fixed).txt

$N_c$  (The Average Number of Customers in Queue in Analytic)  $\Rightarrow$  Customers\_In\_Queue(Fixed).txt

## ۲. تمرین دوم

در این تمرین، شیوه سرویس‌دهی تغییر کرده است. تمرین قبلی تنها مشتری که در اولین خانه صف بود سرویس دریافت می‌کرد و زمان اخراجش از صف می‌رسید اخراج نمی‌شد، اما در این تمرین سرویس‌دهی به صورت *Round Robin* انجام شده و تمامی مشتری‌های حاضر در صف سرویس دریافت می‌کنند و اگر زمان اخراج مشتری‌های حاضر در صف برسد و سرویسشان کامل دریافت نشده باشد از صف اخراج می‌شوند. در پیاده‌سازی این تمرین تغییراتی نسبت به تمرین قبلی وجود دارد که به بیان آن می‌پردازیم. در این تمرین پارامترهای افزوده شده است.

۱.  $Prev - Time$ : زمان رخداد قبلی را نمایش می‌دهد.

۲.  $Free - Time$ : آرایه که نشان می‌دهد چه مقدار از سرویس هر مشتری داخل صف باقی مانده است.

۳.  $Number - of - Customer - in - Queue$ : این پارامتر تعداد مشتری‌های حاضر در صف را در بازه زمانی میان رخداد قبلی و فعلی را نشان می‌دهد.

در این تمرین دارای دو رخداد رسیدن زمان ورود یک مشتری به صف و رسیدن زمان اخراج یک مشتری است. برای آنکه متوجه شویم سرویس یک مشتری تمام شده است یا نه باید  $Free - Time$  آن را با رسیدن هر رخداد محاسبه کنیم، که نحوه محاسبه آن به صورت زیر است.

```
for k in range( 0 , len(Free_Time)):
```

```
    if Free_Time[k] != 'Empty' :
```

```
        Free_Time[k] = Free_Time[k] - (( 1 / Number_of_Customer_in_Queue) * (Time - Prev_Time))
```

حال با هر رخدادی بررسی می‌شود که اگر  $Free - Time$  هر مشتری صفر و یا منفی شد باید به عنوان یک مشتری که سرویس

گرفته صف را ترک کند و یا اگر زمان اخراج آن رسیده و سرویش کامل نشده از صف اخراج شود. تغییراتی که نسبت به تمرین اول انجام شد آن است که بجای یک رخداد رسیدن زمان ورود یک مشتری، رخداد رسیدن زمان اخراج مشتری از صف نیز اضافه شده است. همچنین محاسبه زمان سرویس گرفتن مشتری‌های صف نیز بر اساس قاعده جدید تغییر کرده است.