

ارزیابی کارایی سیستم های کامپیوتری  
نیم سال اول ۱۴۰۰



دانشکده ی مهندسی کامپیوتر

راهنمای انجام تمرین سوم

مدرس : دکتر موقر

❖ آنچه در این تمرین باید تحول دهید :

برای این تمرین شما باید اولاً کد برنامه ی شبیه سازی را به همراه یک Make File که اجرا کننده ی برنامه ی شما است، تحویل دهید. نحوه ی ایجاد Make File برای سیستم عامل لینوکس در انتهای راهنما آمده است. برنامه ی شما باید مقادیر احتمال بلوکه شدن (Pb) و احتمال خارج شدن از صف یا همان ددلاین (Pd) مشتریان را فقط با روش شبیه سازی و به ازای مقادیر مشخص برای پارامترهای  $\mu$  و  $\theta$  که از فایل parameters.conf می خواند و مقادیر ۵ و ۱۰ و ۱۵ برای  $\lambda$  در دو حالت تتا زمان انتظار ثابت و نمایی محاسبه کند و در فایل متنی خروجی چاپ کند. همچنین شما باید دو فایل Excel که حاوی داده های به دست آمده در مورد Pb و Pd مشتریان در روش شبیه سازی به ازای تتای ثابت و تتای نمایی می باشد را تحویل دهید.

❖ آنچه در این تمرین باید به دست آورید :

برای این که بتوانید مقادیر مورد نیاز را برای آن دو فایل Excel که در بالا ذکر شد به دست آورید، لازم است برنامه ای که شما برای خودتان اجرا می کنید Pb و Pd را به ازای  $\lambda$  از ۰.۱ تا ۲۰.۰ و با پرش ۰.۱ محاسبه کند و مقادیر حاصل از روش شبیه سازی را به ازای تتای ثابت و تتای نمایی در فایل های متنی خروجی (ترجیحا به شکل ستونی) چاپ کند تا شما بتوانید این مقادیر را در ستون های مربوطه در فایل Excel کپی کنید و آن فایل ها را تکمیل نمایید.



## ارزیابی کارایی سیستم های کامپیوتری

نیم سال اول ۱۴۰۰

دانشکده ی مهندسی کامپیوتر

راهنمای انجام تمرین سوم

مدرس : دکتر موقر

### ❖ ساختار برنامه (قسمت شبیه سازی) :

برای انجام شبیه سازی، لازم است که شما ابتدا برای مشتریان سیستم و نیز رخدادهای موجود در سیستم ، نوع داده تعریف کنید. نوع داده ی مشتری باید شامل صفاتی از جمله زمان ورود ، زمان انتظار ، زمان سرویس، یک شماره برای مشتری ، زمان سرویس باقی مانده و "کلاس مشتری" (مخصوص این تمرین) باشد . کلاس مشتری را می توانید هنگام ایجاد هر مشتری ، با تولید یک عدد تصادفی بین 0 تا 1 و بررسی کوچکتر یا بزرگتر بودن آن از 0.5 و اختصاص کلاس ۱ یا ۲ به آن مشتری ، انجام دهید (کلاس ۲ مشتریانی هستند که سهم ۲ برابر می برند) .

نوع داده ی رخداد نیز باید شامل صفاتی از جمله نوع رخداد (۲ نوع : ورود مشتری و خارج شدن مشتری از صف یا همان ددلاین - عدم نیاز به تعریف رخداد اتمام سرویس در این تمرین)، زمان وقوع رخداد و نیز شماره ی مشتری مربوط به آن رخداد باشد.

سپس لازم است که شما لیست مشتریان را تعریف کنید. برای این منظور در یک حلقه ی تکرار که به تعداد مشتریان شما تکرار می شود ، هر مشتری را در یک خانه از یک ساختمان داده از نوع داده ی مشتری تعریف می کنید. می توانید زمان ورود اولین مشتری را 0 در نظر بگیرید سپس برای هریک از مشتریان بعدی، یک عدد تصادفی با توزیع نمایی بر اساس فرمول زیر تولید کنید و با زمان ورود مشتری قبلی جمع کنید و به عنوان زمان ورود مشتری فعلی در نظر بگیرید. همچنین برای تعریف زمان سرویس هر مشتری، می توانید از فرمول زیر استفاده کنید با این تفاوت که به جای  $\lambda$  ، پارامتر  $\mu$  را قرار دهید تا عددی تصادفی با توزیع نمایی برای شما تولید شود. واضح است که در ابتدای کار که مشتری سرویسی نگرفته، مقدار اولیه ی صفت "زمان سرویس باقی مانده" نیز برابر با زمان سرویس مشتری است. برای تولید زمان انتظار مشتریان نیز برای حالت تتای ثابت، زمان انتظار تمامی مشتریان برابر  $\theta$  خواهد بود و برای حالت تتای نمایی می توانید برای تولید عدد تصادفی با توزیع نمایی به عنوان زمان انتظار، از فرمول زیر استفاده کنید با این تفاوت که به جای پارامتر  $\lambda$  ، مقدار  $1/\theta$  را قرار می دهید (زیرا تتا در اصل میانگین زمان انتظار مشتریان است و می دانیم که  $rate = 1/average$ ). نحوه ی مقداردهی به فیلد "کلاس مشتری" نیز در بالا ذکر شده است .

$$y = - \frac{\ln(1 - x)}{\lambda}$$



## ارزیابی کارایی سیستم های کامپیوتری

نیم سال اول ۱۴۰۰

دانشکده ی مهندسی کامپیوتر

راهنمای انجام تمرین سوم

مدرس : دکتر موقر

همچنین لازم است که یک ساختمان داده از نوع داده ی مشتری به سبب ۱۲ برای شبیه سازی صف ایجاد کنیم که در این تمرین ، به ازای هر بازه ی زمانی مشخص ، همه ی کسانی که در آن مدت در صف هستند ، با سهمی که بستگی به کلاسشان دارد ، در آن بازه زمانی سرویس خواهد گرفت . همچنین لازم است که در هر لحظه ، صف ۱۲ نفره ی مشتریان به ترتیب صعودی زمان سرویس باقی مانده ی مشتریان مرتب باشد که این موضوع را باید هنگام درج مشتری جدید در صف ۱۲ نفره لحاظ کرد ولی برای راحتی کاری که در ادامه پیش رو دارید ، بهتر است هنگام مرتب کردن مشتریان در صف ، شماره کلاس آنها را هم دخیل کنید یعنی معیار مرتب کردن مشتریان در صف را به جای زمان سرویس باقی مانده ، "زمان سرویس باقی مانده تقسیم بر شماره ی کلاس" در نظر بگیرید و بر این اساس مشتریان را به ترتیب صعودی مرتب کنید زیرا آنچه که در زودتر سرویس گرفتن مشتریان دخیل است ، صرفاً زمان سرویس باقی مانده ی مشتریان نیست بلکه شماره کلاس مشتریان نیز در این موضوع دخیل است و شما با این عمل به نوعی مشتریان را به ترتیب "زمان سرویس باقی مانده ی نسبی" مرتب کرده اید (توجه کنید که در این عملیات صرفاً مشتریان را بر اساس زمان سرویس باقی مانده ی نسبی شان مرتب کرده ایم و تغییری در مقدار زمان سرویس باقی مانده ی آنها نمی دهیم) . برای تعیین طول صف نیز می توان متغیری مجزا تعریف نمود. ضمناً به ۲ متغیر سراسری برای تعیین تعداد مشتریان بلوکه شده و ددلاین شده نیاز داریم که مقدار آنها ابتدا صفر است.

مرحله ی بعدی تعریف لیست رخدادهای است. برای این منظور شما دو گزینه پیش رو دارید . اول آن که در ابتدای کار و پیش از شروع حلقه ی تکرار اصلی برنامه، برای همه ی مشتریانی که تعریف کرده اید، رخداد ورودشان را تعریف کنید و در لیست رخدادهای که یک ساختمان داده از نوع داده ی رخداد است قرار دهید که البته این کار ممکن است به خاطر اشغال حجم حافظه ی RAM شما باعث کندی اجرای برنامه شود. دوم آن که یک شمارنده تعریف کنید و در هر بار اجرای حلقه ی اصلی برنامه، رخداد ورود یک مشتری جدید را ایجاد کنید و به لیست رخدادهای اضافه کنید تا جایی که شمارنده به تعداد مشتریان برسد که البته این کار اندکی پیچیدگی برنامه را بیشتر می کند. توجه کنید که در هر لحظه از برنامه لازم است که لیست رخدادهای به ترتیب صعودی زمان وقوع رخداد مرتب باشد (هنگام درج رخداد جدید در لیست باید به این موضوع توجه کنیم).

سپس به سراغ حلقه ی تکرار اصلی برنامه می رویم. در هر بار اجرای این حلقه، یک رخداد از سر لیست رخدادهای برداشته می شود و بررسی می شود و عمل مربوط به آن رخداد انجام می شود. این حلقه می تواند یک حلقه ی تکرار باشد که پایان آن، زمانی است که لیست رخدادهای خالی شده باشد زیرا پایان برنامه زمانی است که تکلیف همه ی مشتریان روشن شده باشد (یعنی همه آمده اند و یا بلوکه شده اند یا در صف زمان انتظارشان تمام شده و ددلاین شده اند یا سرویس گرفته اند و رفته اند) و این هنگامی رخ می دهد که لیست رخدادهای خالی باشد. همچنین برای استفاده در توابع مختلف و عملیات مختلف برنامه، می توان قبل از شروع حلقه اصلی ، یک متغیر Time برای تعیین زمان در حین اجرای برنامه تعریف کنیم که مقدار اولیه ی آن 0 است و در ابتدای هر تکرار از



## ارزیابی کارایی سیستم های کامپیوتری

نیم سال اول ۱۴۰۰

دانشکده ی مهندسی کامپیوتر

راهنمای انجام تمرین سوم

مدرس : دکتر موقر

حلقه ی اصلی برنامه، مقدار Time برابر با زمان وقوع آن رخدادی می شود که در این تکرار از حلقه قرار است بررسی و اجرا شود. توجه کنید که نیازی به تعریف متغیر از نوع date&time که زمان دنیای واقعی را در حین برنامه نگه دارد نمی باشد .

برای بررسی و اجرای هر رخدادی که در هر تکرار از حلقه ی اصلی می خواهیم بررسی کنیم ، مراحل زیر را انجام می دهیم :

- اگر این رخداد از نوع ورود یک مشتری بود ، ابتدا باید بررسی شود که اندازه ی صف مشتریان چقدر است. اگر طول صف برابر ۱۲ بود یعنی مشتری وارد شده بلوکه می شود و تعداد بلوکه شده ها یکی افزوده می شود. اگر طول صف از صفر تا ۱۱ بود یعنی مشتری می تواند وارد صف شود و در این تمرین هر مشتری ای که وارد صف شود سرویس خواهد گرفت. پس مشتری را در صف جای می دهیم و رخداد مربوط به ددلاین آن مشتری را بر اساس مجموع زمان فعلی (که در متغیر Time هست) و زمان انتظار این مشتری ایجاد می کنیم و در مکان مناسب در لیست رخدادهای قرار می دهیم .
- اگر این رخداد از نوع ددلاین یک مشتری بود ، آن مشتری را از صف حذف می کنیم و طول صف را یکی کم می کنیم و تعداد ددلاین شده ها را یکی اضافه می کنیم و مشتریان بعد از آن را (در صورت وجود) یک واحد به جلو می بریم.

در این تمرین چون به ازای هر بازه ی زمانی خاص تمام مشتریان داخل صف طبق زمانبندی DPS ، بسته به کلاس شان سهمی از کل زمان می برند و به اندازه ی سهم شان سرویس می گیرند، پس در هر تکرار از حلقه ی اصلی ، فاصله ی زمانی بین زمان فعلی (زمان وقوع رخدادی که در این تکرار از حلقه قرار است بررسی شود و در متغیر Time هم ذخیره شده) و زمان وقوع رخداد بعدی (البته به جز آخرین رخداد) را به عنوان یک بازه ی زمانی در نظر می گیریم و بر مقدار " (تعداد مشتریان کلاس ۱ + تعداد مشتریان کلاس ۲ × ۲) " تقسیم می کنیم و برای مشتریان کلاس ۱ به اندازه ی ۱ سهم و برای مشتریان کلاس ۲ به اندازه ی ۲ سهم ، از "زمان سرویس باقی مانده" آنها کم می کنیم. سپس بررسی می کنیم که اگر زمان سرویس باقی مانده ی مشتری ای صفر شده بود یعنی سرویس آن به اتمام رسیده بود، آن را از صف خارج می کنیم و رخداد ددلاین آن را نیز در لیست رخدادهای پیدا می کنیم و حذف می کنیم و طول صف را هم یک واحد کم می کنیم و مشتریان بعد از آن را یک واحد به جلو می بریم . البته این احتمال وجود دارد که برای یکی از مشتریان داخل صف، زمان سرویس باقی مانده اش کمتر از سهمش باشد. بدین منظور چون با روشی که پیش تر گفتیم مشتریان را مرتب می کرده ایم و هر بار از مشتری سر صف (کمترین زمان سرویس باقی مانده ی نسبی) شروع می کنیم و سهم آن مشتری از زمان کل سرویس را می دهیم ، چنانچه زمان سرویس باقی مانده ی آن مشتری کمتر از سهم آن مشتری از زمان کل سرویس در این بازه ی زمانی بود، باقی مانده ی سهم آن مشتری را به مجموع زمان سرویسی که قرار است بین سایر مشتریان تقسیم شود، اضافه می کنیم و برای مشتریان بعدی دوباره مقدار سهم را محاسبه می کنیم تا به اولین مشتری در صف برسیم که زمان سرویس باقی مانده اش بزرگتر یا مساوی سهمش باشد و چیزی اضافه نماند. اگر هم زمان سرویس باقی مانده ی همه ی مشتریان کمتر از سهمشان بود، یعنی بازه ی زمانی بین دو رخداد متوالی از مجموع زمان سرویس مورد نیاز مشتریان داخل



## ارزیابی کارایی سیستم های کامپیوتری

نیم سال اول ۱۴۰۰

دانشکده ی مهندسی کامپیوتر

راهنمای انجام تمرین سوم

مدرس : دکتر موقر

صف در این مدت، کمتر بود، طبیعی است که مازاد آن هدر می رود تا شاید در رخداد بعدی مشتری جدیدی بیاید و سرویس بگیرد. (ممکن است با این الگوریتم تقسیم سهم در سایر دروس نیز آشنا شده باشید).

در پیاده سازی الگوریتم های فوق ، بسته به زبان برنامه نویسی و ساختارهای حلقه و شرط ها و متغیرهای شمارنده ای که استفاده می کنید باید دقت کنید که موارد جزئی از منطق الگوریتم را ناخواسته نقض نکنید . برای مثال موقعی که زمان سرویس باقی مانده ی یک مشتری صفر شد و خواستید آن را از صف خارج کنید و مشتریان بعدی را یک واحد به جلو ببرید، دقت کنید که مشتری بعدی که اکنون جای مشتری خارج شده نشسته است را از قلم نیندازید و نیز کاهش طول صف را در شمارنده ای که مشتریان داخل صف را می شمارد و جلو می رود، لحاظ کنید و یا به عنوان مثال هنگام اختصاص سهم مشتریان از زمان کل سرویس در هر بازه ی زمانی ، به ازای هر مشتری که در صف هست هربار سهم "مشتریان باقی مانده" از "زمان کل باقی مانده" را حساب کنیم و بسته به کلاس ۱ یا ۲ بودن آن مشتری سهمش را از زمان کل باقی مانده بدهیم و به اندازه ای که اکنون به این مشتری سرویس دادیم از زمان کل کم کنیم (اگر به تمام سهمش نیاز نداشت فقط به اندازه ی نیازش کم می کنیم) و برای مشتری بعدی دوباره سهم مشتریان باقی مانده را از زمان کل باقی مانده حساب می کنیم . یعنی نمی توان در ابتدای کار سهم تک تک مشتریان داخل صف را حساب کرد زیرا ممکن است برخی از آنها به مقداری کمتر از سهمشان نیاز داشته باشند که مابقی آن باید به سهم بقیه اضافه شود .

در نهایت زمانی که هیچ رخدادی در لیست رخدادها نمانده بود یعنی تکلیف همه ی مشتریان معلوم شده است و حلقه ی اصلی برنامه پایان می یابد. در این هنگام نسبت تعداد مشتریان بلوکه شده و تعداد مشتریان ددلاین شده به کل مشتریان را به عنوان Pb و Pd در روش شبیه سازی، چاپ می کنیم . می توان کل این برنامه ی نوشته شده را داخل یک حلقه ی تکرار گذاشت که در این حلقه مقدار  $\lambda$  از 0.1 تا 20.0 با پرش 0.1 افزایش می یابد .

### ❖ نحوه ی ایجاد Make File در سیستم عامل های لینوکس :

اگر فرض کنیم که fixed.cpp و exp.cpp دو برنامه ی ما باشند که با زبان C++ نوشته شده اند، با اجرای دستور make در خط فرمان لینوکس، ۲ دستور ابتدایی در تصویر زیر اجرا شده و منجر به ایجاد فایل باینری حاصل از کامپایل این دو برنامه با g++ در لینوکس می شود و با اجرای دستور make run در خط فرمان لینوکس ۲ دستور بعدی در تصویر اجرا شده و باعث اجرای فایل های باینری حاصل و ایجاد خروجی مورد نظر می شود و دستر آخر نیز باعث پاک شدن فایل های باینری می شود.

ارزیابی کارایی سیستم های کامپیوتری  
نیم سال اول ۱۴۰۰



دانشکده ی مهندسی کامپیوتر

راهنمای انجام تمرین سوم

مدرس : دکتر موقر

```
makefile
~/
Open [icon] Save [icon] [icon] [icon] [icon]
output :
    g++ fixed.cpp -o ./fixed.out
    g++ exp.cpp -o ./exp.out
run :
    ./fixed.out
    ./exp.out
clear :
    rm ./*.out
```

با آرزوی موفقیت