



❖ 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 و نیز متوسط تعداد مشتری های داخل صف (**Nc**) را با دو روش شبیه سازی و تحلیلی و به ازای مقادیر مشخص برای پارامترهای μ و θ که از فایل **parameters.conf** می خواند و مقادیر 5 و 10 و 15 برای λ 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** و **Nc** مشتریان در دو روش شبیه سازی و تحلیلی به ازای تتای ثابت و تتای نمایی می باشد را تحویل دهید. در این فایل ها مقادیر خطای مطلق و خطای نسبی بین مقادیر حاصل از روش شبیه سازی و روش تحلیلی محاسبه می شود و به ازای هر روش، سه نمودار رسم می شود. هر قدر خطای شما کمتر باشد و نمودارهای مقادیر حاصل از روش شبیه سازی و تحلیلی بیشتر بر هم انطباق بیشتری داشته باشند، نمره ی بهتری خواهید گرفت.

❖ 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** و **Nc** را به ازای λ از 0.1 تا 20.0 و با پرش 0.1 محاسبه کند و مقادیر حاصل از روش شبیه سازی و تحلیلی مربوط 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.

❖ 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 و یک شماره برای مشتری باشد.



ارزیابی کارایی سیستم های کامپیوتری

نیم سال اول ۱۴۰۱

دانشکده ی مهندسی کامپیوتر

راهنمای انجام تمرین اول

مدرس : دکتر موقر

نوع داده ی رخداد نیز باید شامل صفاتی از جمله نوع رخداد (3 نوع : ورود مشتری، خارج شدن مشتری از صف یا همان ددلاین و خارج شدن مشتری از سرور) ، 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 λ ، پارامتر μ را قرار دهید تا عددی تصادفی با توزیع نمایی برای شما تولید شود . برای تولید زمان انتظار مشتریان نیز برای حالت تتای ثابت، زمان انتظار تمامی مشتریان برابر θ 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 λ ، مقدار $1/\theta$ را قرار می دهید (زیرا تتای میانگین زمان انتظار مشتریان است و می دانیم که $\text{rate} = 1/\text{average}$).

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

همچنین لازم است که یک ساختمان داده از نوع داده ی مشتری به سبب 14 برای شبیه سازی صف ایجاد کنیم که خانه ی شماره یک آن نمایانگر مشتری در حال سرویس گیری است که البته می توان از یک متغیر مجزا نیز به عنوان متغیر سرور برای تعیین مشتری در حال سرویس گیری استفاده کرد. برای تعیین طول صف نیز می توان متغیری مجزا تعریف نمود. ضمناً به 2 متغیر سراسری برای تعیین تعداد مشتریان بلوکه شده و ددلاین شده نیاز داریم که مقدار آنها ابتدا صفر است.

مرحله ی بعدی تعریف لیست رخدادها 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 رخدادها 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 رخدادها 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 رخدادها 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 رخدادها 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 رخدادها 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 رخدادها 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 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. این متغیر در ادامه، می تواند در محاسبه ی متوسط تعداد مشتری های داخل صف نیز به ما کمک کند. 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 سائز صف برابر 14 بود یعنی مشتری وارد شده بلوکه می شود و تعداد بلوکه شده ها یکی افزوده می شود. اگر سائز صف از 1 تا 13 بود یعنی مشتری می تواند وارد صف شود ولی فعلا سرویس نمی گیرد. پس مشتری را در انتهای 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 رخدادها قرار می دهیم. اگر هم سائز صف 0 بود، یعنی مشتری وارد صف و وارد سرور می شود پس رخداد خروج آن مشتری (اتمام سرویس گیری) را بر اساس مجموع زمان فعلی و زمان سرویس این مشتری ایجاد می کنیم و در مکان مناسب در لیست رخدادها قرار می دهیم.
- اگر این رخداد از نوع ددلاین یک مشتری بود، آن مشتری را از صف حذف می کنیم و صف را یک واحد به جلو می بریم و سائز صف را یکی کم می کنیم و تعداد ددلاین شده ها را یکی اضافه می کنیم.
- اگر این رخداد از نوع خروج مشتری از سرور و اتمام سرویس گیری بود، ابتدا آن مشتری را از ابتدای صف مشتریان و از سرور حذف می کنیم و صف را یک واحد به جلو می بریم و اگر نفر دومی در صف بود که اکنون به ابتدای صف رفته، آن را وارد سرور می کنیم و سپس در لیست رخدادهای موجود جستجو می کنیم و رخداد مربوط به ددلاین این مشتری که اکنون وارد سرور شده را می یابیم و حذف می کنیم زیرا طبق شرایط تمرین مشتری ای که وارد سرور شود دیگر ددلاین

ارزیابی کارایی سیستم های کامپیوتری

نیم سال اول ۱۴۰۱



دانشکده ی مهندسی کامپیوتر

راهنمای انجام تمرین اول

مدرس : دکتر موقر

نخواهد شد. همچنین رخداد مربوط به اتمام سرویس گیری و خروج این مشتری از سرور را نیز بر اساس مجموع زمان فعلی و زمان سرویس این مشتری ایجاد و در مکان مناسب از لیست رخداد درج می کنیم.

در هر بار پردازش یک رخداد جدید، تعداد اعضای صف می تواند افزایش یا کاهش یابد. برای محاسبه ی مقدار متوسط تعداد مشتری های داخل صف (NC)، می توانیم از همان متغیر Time استفاده کنیم. لازم است که در ابتدا، یک متغیر سراسری برای نگهداری این مقدار تعیین و هر بار که تعداد افراد حاضر در صف، تغییر می کند، مقدار آن را با توجه به مدت زمانی که از آخرین تغییر گذشته است و نیز تعداد افراد حاضر در صف در طول این مدت، به روزرسانی کنیم.

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، مقدار متوسط تعداد مشتری های داخل صف (NC)، نسبت تعداد مشتری های بلوکه شده به کل مشتری ها (Pb) و تعداد مشتری های ددلاین شده به کل مشتری ها (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 λ از 0.1 تا 20.0 با پرش 0.1 افزایش می یابد.



ارزیابی کارایی سیستم های کامپیوتری

نیم سال اول ۱۴۰۱

دانشکده ی مهندسی کامپیوتر

راهنمای انجام تمرین اول

مدرس : دکتر موقر

❖ ساختار برنامه (قسمت تحلیلی) :

برای محاسبه بخش تحلیلی، ابتدا برای $n=0$ تا $n=14$ ، مقادیر ϕ_1 تا ϕ_{14} را بر اساس فرمول 1 برای حالت تتای نمایی و فرمول 2 برای حالت تتای ثابت (که طرف راست فرمول ها همه مقادیر معلوم هستند) محاسبه می کنیم سپس چون می دانیم مجموع P_0 تا P_{14} برابر 1 است و نیز طبق فرمول 3 می دانیم که هر یک از متغیرهای P_1 تا P_{14} حاصل ضربی از P_0 در مقداری معلوم (که طبق فرمول و براساس یک سری مقادیر ثابت و مقادیر ϕ ها که آنها را قبلا محاسبه کرده ایم ، محاسبه می شود) است ، بنابراین یک معادله داریم به صورت $P_0 + P_0X_1 + \dots + P_0X_{14}=1$ که X ها مقادیر معلومی هستند که قبلا آنها را محاسبه کرده ایم. پس یک معادله و یک مجهول داریم که همان P_0 است. پس P_0 را می یابیم و با قرار دادن آن در هر یک از $P_1=P_0X_1$ تا $P_{14}=P_0X_{14}$ ، مقادیر P ها را برای 1 تا 14 می یابیم که در نهایت P_{14} می شود همان احتمال بلاک ما یعنی P_b و از روی آن بر اساس فرمول 4 ، P_d ، یعنی احتمال ددلاین شدن را هم حساب می کنیم. محاسبه ی N_c نیز مطابق با فرمول 5 صورت می گیرد.

$$\varphi_n(\mu) = \frac{n!}{\prod_{i=0}^n (\mu + i/\bar{\theta})}$$

فرمول 1

$$\varphi_n(\mu) = \frac{n!}{\mu^{n+1}} (1 - e^{-\mu\bar{\theta}} \sum_{i=0}^{n-1} \frac{(\mu\bar{\theta})^i}{i!})$$

فرمول 2

$$P_n = \begin{cases} P_0 \left(\frac{\lambda}{\mu} \right), & n = 1 \\ P_0 \lambda^n \frac{\varphi_{n-1}(\mu)}{(n-1)!}, & n > 1 \end{cases}$$

فرمول 3



$$P_d + P_b = 1 - \frac{\mu}{\lambda} (\sum_{i=1}^K P_i) = 1 - \frac{\mu}{\lambda} (1 - P_0)$$

فرمول 4

$$N_c = \sum_{i=1}^K (i \cdot P_i)$$

فرمول 5

❖ 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، 2 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 2 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.

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