



هدف از انجام این پروژه آشنایی فراخوانی‌های سیستمی زبان C و یادگیری مبانی socket programming است.

شرح تمرین

در این پروژه می‌خواهیم یک سیستم سفارش غذا تحت خط فرمان فراهم کنیم که با استفاده از socket programming و فراخوانی‌های سیستمی زبان C انجام می‌شود. توجه کنید که این سیستم دارای ادمین مرکزی نمی‌باشد (centralized نیست) و تمام نودها باید بتوانند با هم بدون ادمین مرکزی ارتباط برقرار کنند و با هم با استفاده از سوکت ارتباط برقرار کنند. همچنین بخش مهمی از پروژه مربوط به بررسی تمامی خطاهای احتمالی در برنامه می‌باشد. آنها را به طور دلخواه هندل کنید و پیام‌های دلخواه را نمایش دهید.

سوکت چیست؟

سوکت یک مکانیزم برای برقرای ارتباط بین دو پردازنده بین یک یا چند ماشین است. در این ارتباط که دو طرفه است، سوکت مانند یک پایانه عمل می‌کند که ما یا اطلاعات را به آن می‌فرستیم یا از آن دریافت می‌کنیم. در اینجا سوکت یک انتزاع از لایه‌های پایین‌تر سیستم عامل است که این ارتباط را میسر می‌کند.

فهرست دستوریخت غذاها

لیست تمام دستوریخت غذاها به صورت یک فایل recipes.json در اختیار شما قرار خواهد گرفت که در آن مواد اولیه مورد نیاز به همراه مقدار آنها مشخص شده است.

وارد سامانه‌شدن

در سامانه سه نقش خواهیم داشت: ۱- صاحب رستوران ۲- تامین‌کننده ۳- مشتری .

صاحب رستوران مواد اولیه مورد نیاز خود را از تامین‌کننده دریافت می‌کند و به سفارش‌های مشتریان پاسخ می‌دهد. همچنین هر کدام از نقش‌ها تعدادی دستور دارند که مخصوص خودشان خواهد بود.

هر شخصی با هر عنوان که وارد سامانه می‌شود، ابتدا نام خود را وارد می‌کند. همچنین پورتهای عنوان آرگومان ورودی دریافت می‌کند که می‌تواند روی آن به پیام‌های دیگر افراد در سامانه گوش بدهد و وضعیت آنها را دنبال کند. دقت شود که این کار نیازمند یک پدازه جدا می‌باشد. هر شخص که وارد سیستم می‌شود و username خود را وارد می‌کند، این موضوع و اضافه شدنش به سیستم را از طریق ارتباط udp روی پورت وارد شده به عنوان آرگومان broadcast می‌کند. توجه کنید که برای اطلاع از وضعیت هر نود از همین پورت و از طریق ارتباط udp اطلاعات منتشر می‌شود.

دقت شود که نام هر فرد در سامانه یکتا می‌باشد و نباید تکراری باشد (برای این کار نمی‌توانید از یک فایل مشترک استفاده کنید و اسامی را در آنجا بنویسید).

حال برنامه‌های مربوط به مشتری، رستوران و تامین‌کننده بدین شکل اجرا می‌شوند (ابتدا اجرا کردن فایل اجرایی برنامه و به عنوان آرگومان، پورتهای که روی آن ارتباط udp انجام می‌شود یا به اصطلاح (server_port

Customer1	Restaurant1	Supplier1
<pre>→ Customer ./customer.out 8001 Please enter your username : amirali wellcome amirali as customer!! >></pre>	<pre>→ Restaurant ./restaurant.out 8001 Please enter your username : reza wellcome reza as Restaurant!! >></pre>	<pre>→ Supplier ./supplier.out 8001 Please enter your username : saman wellcome saman as Supplier!! >></pre>

تایمر

در چند دستور لازم است که مدتی منتظر پاسخ تامین‌کننده یا صاحب رستوران باشیم و اگر پاسخی دریافت نشد، باید از حالت انتظار خارج شویم و درخواست پایان می‌یابد. برای پیاده‌سازی این بخش باید از سیگنال SIGALRM و سیستم‌کال alarm استفاده کنیم.

همزمانی سیستم

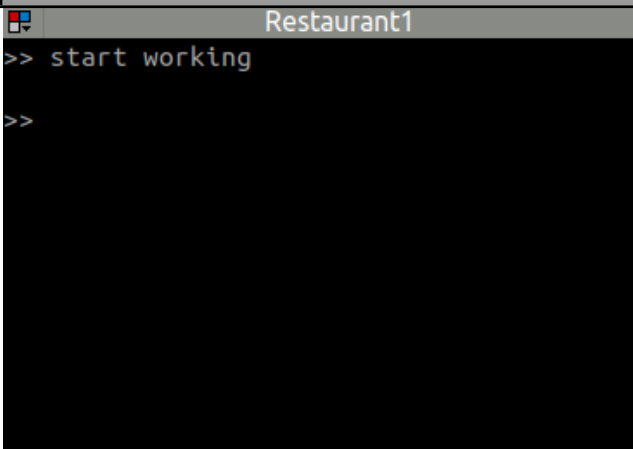
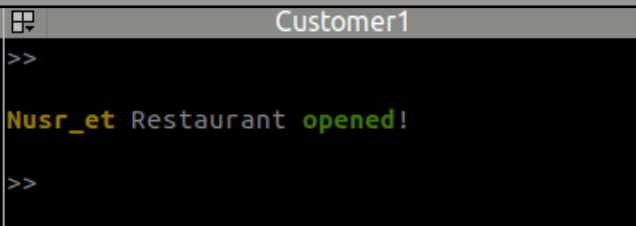
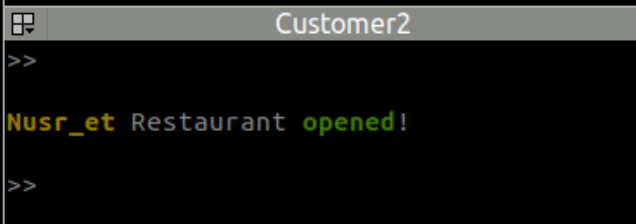
در کل طول اجرای برنامه، صاحب رستوران می‌تواند به درخواست‌های مختلف را داشته باشد و به هر کدام پاسخ بدهد. همچنین می‌تواند در حالی که چند درخواست دارد، خود درخواست مواد اولیه را از تامین‌کننده‌ها داشته باشد. با این حال برخی سیستم‌کال‌ها حالت blocking دارند و اجرای برنامه

در آن قسمت‌ها متوقف می‌شود. برای حل مشکل لازم است از سیستم‌کال select استفاده شود. این سیستم‌کال می‌تواند ارتباطات و I/O-ها را بدون بلاک‌شدن مدیریت کند. در این پروژه هم باید به کمک این سیستم‌کال، تمام I/O-ها را مدیریت کنید، بدون اینکه روند اجرای برنامه متوقف شود. همچنین توجه کنید که با قراردادن stdin در لیستی که به select می‌دهید می‌توانید بدون بلاک شدن از کنسول ورودی را بخوانید.

مجموعه دستورات صاحب رستوران

باز کردن رستوران

با زدن این دستور رستوران شروع به کار می‌کند و مشتریان می‌توانند از آن غذا سفارش دهند. تمام دستورات صاحب رستوران در صورتی قابل انجام است که رستوران باز باشد. همچنین توجه کنید هنگامی که رستوران باز می‌شود لازم است تا در یک پیامی این اتفاق روی پورت udp گفته شده broadcast شود و لازم است به صورت یک پیام برای مشتریان نمایش داده شود. هنگامی که صاحب رستوران وارد سیستم می‌شود نیز به طور خودکار رستوران، باز حساب می‌شود و اینکه این موضوع به صورت یک پیام به مشتریان نمایش داده شود به عهده خودتان است.

دستور	
Restaurant -> command -> start working Customer -> notify -> <name> Restaurant opened!	
مثال	
 <pre>>> start working >></pre>	 <pre>>> Nusr_et Restaurant opened! >></pre>  <pre>>> Nusr_et Restaurant opened! >></pre>

بستن رستوران

با زدن این دستور رستوران بسته می‌شود و تا زمانی که رستوران باز بشود، درخواست‌های زده شده به رستوران رد می‌شوند. صاحب رستوران هم هیچ کدام از دستورات رستوران به جز باز کردن رستوران را نمی‌تواند اجرا کند. دقت کنید که در صورتی رستوران می‌تواند بسته شود که به تمام درخواست‌های سفارش داده شده پاسخ داده باشد.

دستور	
Restaurant -> command -> break Customer -> notify -> <name> Restaurant closed!	
مثال	
 Restaurant1	 Customer1
>> break	>>
>>	Nusr_et Restaurant closed!
	>>
	 Customer2
	>>
	Nusr_et Restaurant closed!
	>>

مشاهده فهرست مواد اولیه

با زدن این دستور مواد اولیه باقی‌مانده موجود در انبار نمایش داده می‌شود. دقت کنید که اگر موجودی یک ماده اولیه صفر باشد، لازم به نمایش آن نیست.

دستور
Restaurant -> command -> show ingredients

مثال	
Restaurant1	Restaurant2
<pre>>> show ingredients ----- ingredient/amount Onion 50 Saffron 40 Saffron 40 Sugar 200 Garlic 160 ----- >></pre>	<pre>>> show ingredients ----- ingredient/amount Turmeric 100 Eggs 5 Stew_beef 150 ----- >></pre>

مشاهده فهرست دستورپخت‌ها

با زدن این دستور فهرستی از دستورپخت‌ها نمایش داده می‌شود که در آن مواد اولیه غذاها نیز نمایش داده می‌شود. همان طور که گفته شد تمامی این دستورپخت‌ها در یک فایل نوشته شده و لازم است که در همان شروع این فایل توسط برنامه صاحب رستوران خوانده شود.

دستور	
Restaurant -> command -> show recipes Restaurant -> notify -> list of recipes (custom format)	
مثال	
Restaurant1	Restaurant2
<pre>>> show recipes ----- 1- Fesenjan: Onions : 100 Sugar : 70 Turmeric : 200 2- Tahchin: Saffron : 10 Eggs : 2 3- Baghali_Polo: Dill : 50 Saffron : 5 Turmeric : 500 ----- >></pre>	<pre>>> show recipes ----- 1- Fesenjan: Onions : 100 Sugar : 70 Turmeric : 200 2- Tahchin: Saffron : 10 Eggs : 2 3- Baghali_Polo: Dill : 50 Saffron : 5 Turmeric : 500 ----- >></pre>

مشاهده فهرست تامین‌کننده‌ها

با زدن این دستور لیست تامین‌کننده‌های موجود در سامانه با پورت که روی آن ارتباط tcp است نمایش داده می‌شود تا صاحب رستوران بتواند مواد اولیه خود را از تامین‌کننده دریافت بکند. توجه کنید که با هر ترتیبی از وارد شدن صاحب رستوران‌ها و تامین‌کننده‌ها این دستور باید به درستی کار کند و لیست تامین‌کننده‌ها تا جای ممکن به طور کامل گفته شود (همچنین نباید نام آنها در یک فایل نوشته شود و به طور همزمان از آن فایل اسامی را بخوانند).

دستور	
Restaurant -> command -> show suppliers Restaurant -> notify -> list of suppliers (custom format)	
مثال	
Restaurant1	Restaurant2
<pre>>> show suppliers ----- username/port Sysco_Corporation 3233 Jerónimo_Martins 3231 Bunzl_plc 3250 US_Foods_Holding_Corp 3256 ----- >></pre>	<pre>>> show suppliers ----- username/port Sysco_Corporation 3233 Jerónimo_Martins 3231 Bunzl_plc 3250 US_Foods_Holding_Corp 3256 ----- >></pre>

درخواست مواد غذایی

با زدن این دستور صاحب رستوران می‌تواند مواد اولیه مورد نیاز خود را از تامین‌کننده مورد نظر دریافت بکند. به طوری که با وارد کردن پورت مخصوص تامین‌کننده که روی آن ارتباط tcp است و کالای مورد نیاز خود و همچنین مقدار آن درخواست او انجام می‌شود. هنگامی که درخواست انجام شد و صاحب رستوران منتظر پاسخ تامین‌کننده بود، کامندلاین مربوط به صاحب رستوران لاگ می‌شود و نمی‌تواند چیزی تایپ کند تا زمانی که تامین‌کننده پاسخ او را بدهد. در صورتی که پاسخ مثبت بود، مواد اولیه مورد نیاز اضافه می‌شود و در غیر این صورت پیغام

مناسب داده می‌شود. در این زمان یک پیام به تامین‌کننده نمایش داده می‌شود که درخواست جدیدی برای او آمده است. تا 90 ثانیه اگر تامین‌کننده پاسخ صاحب رستوران را نداد باید پیام timeout به صاحب رستوران نمایش داده شود.

دستور	
Restaurant -> command -> request ingredient Restaurant -> input -> port of supplier Restaurant -> input -> name of ingredient Restaurant -> input -> number of ingredient Restaurant -> waiting for the supplier's response ... Supplier -> notify -> new request ingredient! Supplier -> response -> response to request ingredient (yes or no) Restaurant -> notify -> (<name> Supplier accepted!) or (<name> Supplier denied!) or (Time out!)	
مثال	
Restaurant1	Supplier1
<pre>>> request ingredient -> port of supplier: 3233 -> name of ingredient: Saffron -> number of ingredient: 20 waiting for the supplier's response ... Sysco_Corporation Supplier accepted! >></pre>	<pre>>> new request ingredient! >> answer request -> your answer (yes/no): yes >></pre>

مشاهده فهرست درخواست‌ها

با زدن این دستور صاحب رستوران می‌تواند لیست درخواست‌هایی که پاسخ نداده و منتظر جواب او هستند را مشاهده کند. برای این منظور لازم است پورت دارای ارتباط tcp که مشتری روی آن منتظر پاسخ صاحب رستوران است، نام مشتری و نام غذای مدنظر او نمایش داده شود. دقت شود که برای هر درخواست ۳ سه حالت وجود دارد: ۱- پذیرفته می‌شود ۲- رد می‌شود ۳- زمان آن تمام می‌شود و عملاً رد می‌شود.

توجه کنید که در صورتی که هر کدام از این درخواست‌ها خاتمه یافت باید از این لیست خارج شود (پاسخ مثبت یا منفی به درخواست داده شود یا زمان آن تمام شود).

دستور	
Restaurant -> command -> show requests list Restaurant -> notify -> list of requests (custom format)	
مثال	
Restaurant1	Restaurant2
<pre>>> show requests list ----- username/port/order Misagh 5099 Cordon_bleu Hamed 5111 Chelo_Kebab Nooghani 5211 Shole_mashhadi Hoda 5216 Jigar ----- >></pre>	<pre>>> show requests list ----- username/port/order Pouya 1129 Ferapeh Saman 4410 Baghali_Polo Padi 4449 Ghaza_regimi Hadi 4900 Fesenjan ----- >></pre>

پاسخ به درخواست‌های مشتریان

با زدن این دستور صاحب رستوران می‌تواند به درخواست‌های سفارش مشتریان پاسخ بدهد. بدین صورت که با وارد کردن پورت دارای ارتباط tcp که مشتری روی آن منتظر جواب رستوران است و پاسخ بله یا خیر به آن درخواست پاسخ داده می‌شود و سپس به مشتری اطلاع داده می‌شود. توجه کنید که خطاهای احتمالی مانند نبودن مواد اولیه کافی را باید به شکل دلخواه، هندل و پیاده‌سازی کنید.

دستور
Restaurant -> command -> answer request Restaurant -> input -> port of request Restaurant -> input -> (yes) or (no) (if restaurant has not ingredients, yes leads to error and denied is sent to the customer) Customer -> notify -> (<name> Restaurant accepted and your food is ready !) or (<name> Restaurant denied and cry about it!) or (Time out !)

مثال	
Customer1	Restaurant1
<pre>>> order food -> name of food: Shole_mashhadi -> port of restaurant: 2020 waiting for the restaurant's response ... Nusr_et Restaurant accepted and your food is ready! >></pre>	<pre>>> new order! >> answer request -> port of request: 5211 -> your answer (yes/no): yes >></pre>

مشاهده تاریخچه فروش‌ها

با زدن این دستور صاحب رستوران می‌تواند تمام سفارش‌هایی که در گذشته داشته و در حال حاضر در لیست درخواست‌ها نیستند را به همراه وضعیت نهایی آن درخواست‌ها مشاهده کند. در آن باید نام مشتری سفارش‌دهنده و سفارش او و استیت نهایی درخواست نمایش داده‌شود.

دستور	
Restaurant -> command -> show sales history Restaurant -> notify -> list of sales (custom format)	
مثال	
<pre>----- username/order/result Dante Ferapeh accepted Virgil Jigar denied ----- >></pre>	<pre>----- username/order/result Jak Ferapeh accepted Daxter Jigar accepted ----- >></pre>

مجموعه دستورات تامین کننده

پاسخ به درخواست های رستوران ها

با زدن این دستور، تامین کننده می تواند به درخواست مواد غذایی پاسخ بدهد. توجه کنید که تامین کننده دارای لیست افراد درخواست دهنده نیست و در یک زمان فقط یک رستوران می تواند منتظر پاسخ تامین کننده باشد.

در این قسمت خطاهای احتمالی را به طور دلخواه هندل کنید. مثلاً اگر درخواستی برای تامین کننده وجود نداشت و تامین کننده وارد این قسمت شد باید متن خطای مناسب نمایش داده شود. همان طور که گفته شد، تامین کننده 90 ثانیه فرصت دارد به درخواست پاسخ دهد. در غیر این صورت درخواست منقضی می شود و درخواستی منتظر پاسخ نداریم.

دستور
Supplier -> command -> answer request Supplier -> input -> (yes) or (no) Restaurant -> notify -> (<name> Supplier accepted !) or (<name> Supplier denied !) or (Time out !)
مثال
مانند درخواست مواد غذایی

مجموعه دستورات مشتری

مشاهده فهرست رستوران ها

با زدن این دستور، مشتریان می توانند فهرستی از رستوران ها به همراه پورت دارای ارتباط tcp آنها را مشاهده کنند.

توجه کنید که ترتیب ورود مشتریان و رستوران ها به سامانه نباید تاثیری در خروجی این قسمت داشته باشد (همچنین نباید نام آنها در یک فایل نوشته شود و به طور همزمان از آن فایل اسامی را بخوانند).

دستور
Customer -> command -> show restaurants

Customer -> notify -> list of restaurants (custom format)	
مثال	
<div> <div>Customer1</div> <pre>>> show restaurants ----- username/port Nusr_et 2020 CZN 2024 McDonald 2031 ----- >></pre> </div>	<div> <div>Customer2</div> <pre>>> show restaurants ----- username/port Nusr_et 2020 CZN 2024 McDonald 2031 ----- >></pre> </div>

مشاهده فهرست غذاها

با زدن این دستور مشتریان می‌توانند اسامی غذاها را مشاهده کنند که از همان لیست دستور پخت‌ها قابل استنتاج است. دقت کنید که منبع این اسامی همان فایل مربوط به دستوریست می‌باشد.

دستور	
Customer -> command -> show menu Customer -> notify -> menu (custom format)	
مثال	
<div> <div>Customer1</div> <pre>>> show menu ----- 1- Fesenjan 2- Tahchin 3- Baghali_Polo ----- >></pre> </div>	<div> <div>Customer2</div> <pre>>> show menu ----- 1- Fesenjan 2- Tahchin 3- Baghali_Polo ----- >></pre> </div>

سفارش غذا

با زدن این دستور مشتری می‌تواند درخواست خود را در رستوران ثبت کند. بدین صورت که با وارد کردن نام غذا و پورت مربوط به رستوران که روی آن ارتباط tcp است باید منتظر پاسخ رستوران بماند و بعد از گرفتن پاسخ، پیغام مناسب نمایش داده شود.

هنگامی که درخواست انجام شد و مشتری منتظر پاسخ رستوران است، کامندلاین مربوط به مشتری لاگ می‌شود و نمی‌تواند چیزی تایپ کند تا زمانی که رستوران پاسخ او را بدهد.

تا 120 ثانیه اگر صاحب رستوران پاسخ مشتری را نداد باید پیام timeout به مشتری نمایش داده شود.

توجه کنید که بخش مهمی از نمره شما مربوط به پیاده‌سازی این دستور به گونه‌ای که مشتریان مختلف بتوانند به طور همزمان از یک رستوران درخواست بدهند می‌باشد.

دستور
Customer -> command -> order food Customer -> input -> name of food Customer -> input -> port of restaurant Customer -> waiting for the restaurant's response Restaurant -> notify -> new order! Restaurant -> response -> response to order (yes or no) Customer -> notify -> (<name> Restaurant accepted and your food is ready!) or (<name> Restaurant denied and cry about it!) or (Time out!)
مثال
مانند پاسخ به درخواست‌های مشتریان

نکات و نحوه تحویل

- در این پروژه کدهای شما باید با زبان c باشد که با gcc قابل کامپایل باشد.

- توجه کنید که مثال‌های داده‌شده صرفاً مثالی از قالب خروجی که انتظار داریم می‌باشد و به شکل دلخواه می‌توانید آنها را نمایش دهید. توجه کنید که خروجی دارای اطلاعات لازمه و درست باشد.
 - تمیزی کد بخش مهمی از نمره شما را شامل می‌شود.
 - در حین اجرای برنامه باید از log-های مناسبی استفاده کنید که روند اجرای برنامه به طور کامل مشخص باشد. مثلاً کاربر x وارد سامانه شد یا درخواست y ثبت شد و
 - پیاده‌سازی شما باید به کمک سیستم‌کال‌هایی مانند open, write, read, ... باشد و استفاده از کتابخانه‌ها و توابعی مانند fopen مجاز نیست و نمره منفی به آن تعلق می‌گیرد. این توابع در این [لینک](#) قابل مشاهده هستند.
 - توابع کتابخانه‌ای که توسط سیستم‌کال‌ها قابل پیاده‌سازی نیستند مانند sprintf, strcpy, strcat, atoi, ... مجاز هستند.
 - برای آشنایی با برنامه نویسی سوکت می‌توانید از منابع زیر استفاده کنید. همچنین سوالات خود را می‌توانید در جلسات رفع اشکال مطرح کنید.
- <https://beej.us/guide/bgnet/html/#client-server-background>
- <https://beej.us/guide/bgnet/html/#system-calls-or-bust>
- <https://beej.us/guide/bgnet/html/#broadcast-packetshello-world>
- فایل نهایی که تحویل می‌دهید باید شامل کدهای توسعه داده‌شده (ترجیحاً به همراه Makefile) باشد. در نهایت باید سه فایل اجرایی restaurant.out, supplier.out و customer.out خروجی داده شود. کدهای خودتان را در قالب یک فایل فشرده به نام OS_CA1_<SID>.zip در صفحه‌ی درس آپلود کنید.
 - در صورت داشتن سوال می‌توانید از طریق فروم درس یا شرکت در جلسات رفع اشکال سوالات خود را مطرح کنید.

موفق باشید!