شبیه‌سازی تعامل بین مشتریان، صرافی‌های ارز دیجیتال و بانک با استفاده از پروتکل UDP در زبان برنامه‌نویسی C++

مقدمه

در این پروژه، هدف طراحی و پیاده‌سازی سیستمی است که تعاملات بین مشتریان، صرافی‌های ارز دیجیتال و یک بانک را در بستر شبکه شبیه‌سازی کند. این سیستم با استفاده از پروتکل UDP برای ارتباطات بین موجودیت‌ها پیاده‌سازی خواهد شد و از زبان برنامه‌نویسی C++ بدون بهره‌گیری از کتابخانه‌های خارجی استفاده می‌کند. هر موجودیت دارای یک پورت اختصاصی است و از طریق آن به تبادل پیام با سایر موجودیت‌ها می‌پردازد.

تعریف موجودیت‌ها

سیستم از سه موجودیت اصلی تشکیل شده است:

۱. مشتری(Client)
۲. صرافی(Exchange)
۳. بانک(Bank)

هر یک از این موجودیت‌ها وظایف و قابلیت‌های خاص خود را دارند که در ادامه به تفصیل شرح داده می‌شود.

هر موجودیت پس از عضویت در سیستم ، بر روی یک پورت مشترک میان تمامی موجودیت ها ، خود را معرفی می کند و پورت مشخص شده خود را اعلام می نماید. پس از آن تمامی موجودیت ها یک درخواست به او ارائه می کنند و پورت مشخص خود را به او ارائه می دهند. همچنین تمامی موجودیت ها باید در لحظه ای که یک موجودیت جدید وارد میشود ، باید پورت مشخص او را جهت استفاده ذخیره نمایند و پورت مشخص خود را به او اعلام نمایند. بانک اولین موجودیتی است که عضویت خود را اعلام می کند و دارای پورت ثابت ۹۹۹۹ می باشد.

شرح عملکرد موجودیت‌ها

۱ .موجودیت مشتری (Client) :

مشتری به عنوان کاربر سیستم، می‌تواند از خدمات زیر استفاده کند:

استعلام قیمت ارز دیجیتال: مشتری می‌تواند از یک صرافی خاص قیمت یک ارز دیجیتال مشخص را استعلام کند. پیام درخواست قیمت به پورت صرافی ارسال می‌شود و پاسخ در قالب قیمت فعلی ارز بازگشت داده می‌شود.

دریافت لیست صرافی ها: مشتری می تواند از این طریق لیست تمامی صرافی ها را دریافت کند. مسئول پاسخگویی به این درخواست ، موجودیت بانک خواهد بود و لیست صرافی ها را به مشتری ارائه می کند.

خرید ارز دیجیتال: مشتری می‌تواند درخواست خرید یک ارز دیجیتال را به صرافی ارسال کند. پیش از خرید، سیستم باید بررسی کند که نقدینگی کافی در کیف پول مشتری موجود باشد. پس از تأیید خرید، میزان ارز در کیف پول مشتری افزایش یافته و نقدینگی او کاهش می‌یابد.

فروش ارز دیجیتال: مشتری می‌تواند درخواست فروش ارز دیجیتال را به صرافی ارسال کند. سیستم بررسی می‌کند که میزان ارز مورد نظر در کیف پول مشتری موجود باشد. همچنین باید تأیید شود که صرافی نقدینگی کافی برای خرید ارز از مشتری دارد.

مشـاهده موجودی کیف پول: مشـتری می‌تواند موجودی کیف پول خود شـامل میزان نقدینگی و ارزهای دیجیتال خریداری‌شـده را مشـاهده کند. در شروع سیستم، موجودی نقدینگی و ارزهای مشتری برابر صفر است. برای میزان نقدینگی باید یک درخواست به بانک ارائه کند تا میزان موجودی فعلی خود را از بانک دریافت نماید.

درخواست افزایش نقدینگی از بانک: مشتری می‌تواند به بانک درخواست افزایش نقدینگی دهد. بانک با بررسی سقف مجاز تأمین نقدینگی، درخواست مشتری را تأیید یا رد می‌کند.

مشاهده سابقه تراکنش‌ها: مشتری می‌تواند سابقه تراکنش‌های خود را مشاهده کند. این سابقه شامل : تغییرات در نقدینگی و خرید و فروش ارزهای دیجیتال می باشد.

۲ .موجودیت صرافی (Exchange) :

صرافی به عنوان واسط خرید و فروش ارز دیجیتال، وظایف زیر را بر عهده دارد:

عرضـه اولیه ارز دیجیتال: صـرافی می‌تواند یک ارز دیجیتال جدید را برای اولین بار عرضـه کند. قیمت عرضـه اولیه بر اسـاس تعداد مشـتریانی که درخواسـت خرید ارسـال کرده‌اند تعیین می‌شـود. صـرافی باید بازه زمانی پیش‌فروش و تعداد ارزهای پیش‌فروش‌شـده را کنترل کند تا از حد مجاز فراتر نرود.

فروش ارز دیجیتال: صـرافی می‌تواند ارزهای موجود خود را به مشـتریان بفروشـد. قیمت فروش ارز باید به‌روزرسـانی شـود. برای این منظور، صـرافی در زمان‌های مشخص از سایر صرافی‌ها قیمت ارز را استعلام می‌گیرد. فرآیند فروش ارز بصورت خودکار رخ می دهد و سیستم باید بصورت خودکار، شرایط لازم برای فروش ارز مانند موجودی مشتری و موجودی ارز خود را بررسی نماید و نیازی به مداخله فردی در سمت صرافی نمی باشد.

خرید ارز دیجیتال: پس از فروش مقادیری ارز و افزایش نقدینگی، صرافی می‌تواند با بررسی بازار، ارزهای دیگری را خریداری و در ذخایر خود نگهداری کند. در این بخش نیز شرایط لازم برای خریداری ارز توسط صرافی نیز باید بصورت خودکار توسط سیستم بررسی شود.

مدیریت نقدینگی: صرافی باید موجودی نقدینگی خود را مدیریت کند تا توانایی خرید ارز از مشتریان را داشته باشد.

۳ .موجودیت بانک (Bank) :

بانک به عنوان تأمین‌کننده نقدینگی، خدمات زیر را ارائه می‌دهد:

نگهداری نقدینگی مشتریان : مشتریان تمام موجودی نقدی خود را در بانک نگهداری میکنند و هر زمان که نیاز داشته باشند ، از بانک دریافت خواهند کرد. همچنین بانک باید هر لحظه به درخواست مشتریان برای میزان موجودی نقدی خود پاسخ بدهد.

تأمین نقدینگی برای مشتریان: مشتریان می‌توانند از بانک درخواست افزایش نقدینگی کنند. بانک باید سقف مجاز تأمین نقدینگی برای هر مشتری را بررسی کرده و در صورت امکان نقدینگی مورد نظر را تأمین کند.

مدیریت سقف نقدینگی: بانک باید میزان نقدینگی تخصیص‌یافته به هر مشتری را ثبت و مدیریت کند تا از سقف مجاز فراتر نرود.

توجه شود که بانک دارای موجودیتی واحد است و در کل سیستم فقط یک موجودیت بانک وجود دارد.

مکانیزم تعیین قیمت ارز در یک صرافی :

هر صـرافی پس از فروش مقدار مشـخصـی از هر ارز موجود (۵ درصـد از موجودی هر ارز) می تواند قیمت آن ارز را ۳ درصـد افزایش دهد. همچنین در صورتی که به میزان ۳ درصد از موجودی فعلی خود را از کاربران خریداری کند ، می تواند قیمت آن ارز را ۲ درصد کاهش دهد. این شیوه باعث میشود تا تعادل قیمتی بین صرافی های مختلف برقرار شود. این مقادیر برای خرید و فروش ارز میان صرافی ها نیز برقرار می باشد و باید توسط برنامه ، بررسی و مدیریت شود.

پیاده سازی پروتکل مناسب جهت تبادل ارز میان دو صرافی بر عهده شما می باشد و باید توجیه مناسبی برای ایده خود داشته باشید.

پروتکل ارتباطی بین موجودیت‌ها

ارتباط بین موجودیت‌ها از طریق پروتکل UDP صورت می‌گیرد. هر موجودیت دارای یک پورت اختصاصی است و از طریق آن پیام‌ها را ارسال و دریافت می‌کند.

قالب پیام‌ها:

پیام‌ها در قالب متن ساده (Plain Text)با فرمت مشخص ارسال می‌شوند. هر پیام شامل موارد زیر است:

- نوع پیام (Request/Response)
- شناسه فرستنده
- شناسه گیرنده
- نوع عملیات (مانند خرید ، فروش و استعلام قیمت)
- پارامترهای مرتبط (مانند نام ارز ، مقدار ارز و میزان نقدینگی

نمونه پیام‌ها:

درخواست قیمت ارز :

REQUEST | ClientID: 1 | ExchangeID: 2 | Operation: GetPrice | Currency: Bitcoin

پاسخ قیمت ارز :

RESPONSE | ExchangeID: 2 | ClientID: 1 | Price: 25000

درخواست افزایش نقدینگی :

REQUEST | ClientID: 1 | BankID: 0 | Operation: AddFunds | Amount: 1000

برای ساختار پیام ها ، طراحی فرمت مناسب به عهده شماست و می توانید بسته به نیاز ، اطلاعات مختلفی را منتقل نمایید. اما توجه کنید که باید در هر پیغام ، تمام اطلاعات مورد نیاز برای توجیه درست بودن فرآیند نمایش داده شود. در هنگام ارائه ، باید درستی عملیات از طریق پیام نمایش داده شده ، ثابت شود و توجیهات شفاهی یا عملکردی برای این موضوع مورد پذیرش نمی باشد.

بخش های امتیازی :

۱. زمان‌بندی به‌روزرسانی قیمت‌ها: پیاده‌سازی یک تایمر برای صرافی‌ها تا در بازه‌های زمانی مشخص قیمت ارزها را از سایر صرافی‌ها استعلام و به‌روزرسانی کنند.

۲. امنیت پیام‌ها: افزودن یک شناسه اعتبارسنجی (Token) به پیام‌ها برای جلوگیری از ارسال پیام‌های جعلی.

برخی نکات اضافی :

- لازم است که پروژه با زبان برنامه نویسی C++ پیاده سازی شود و با g++ قابل کامپایل باشد.
- تمیزی کد بخش مهمی از نمره را شامل می شود.
- در حین اجرای برنامه باید از log های مناسبی استفاده کنید که روند اجرای برنامه به طور کامل مشخص باشد. مثلا کاربر x مقدار y ارز به صرافی z فروخت.
- پیاده سازی شما باید به کمک سیستم کال هایی مانند read ، write ، open و ... باشد و استفاده از کتابخانه ها و توابعی مانند fopen مجاز نیست و نمره منفی به آن تعلق میگیرد. این توابع در این لینک قابل مشاهده هستند.
- توابع کتابخانه ای که توسط سیستم کال ها قابل پیاده سازی نیستند مانند strcpy ، sprint ، atoi ، strcat و ... مجاز هستند.
- برای هر موجودیت باید پورت اختصاصی وجود داشته باشد و نیاز است سیستم این پورت ها را مدیریت نماید. همچنین پیاده سازی مکانیزم هایی برای مدیریت خطا مانند بررسی پیام های نامعتبر یا مفقود شده ضروری است.

- پروژه در قالب تیم های ۲ نفره خواهد بود و نیاز است هر دو نفر تسلط کافی به بخش های مختلف پروژه داشته باشند. هرگونه شباهت غیرمتعارف میان پروژه ها ، موجب از دست رفتن نمره پروژه خواهد شد و این شباهت میان پروژه ها با الگوریتم های موجود بررسی خواهد شد.
- ممکن است در برخی از قسمت های پروژه ، جزئیاتی توصیف نشده باشند و پیاده سازی این بخش ها به عهده شما قرار داده شده است اما باید برای تمام بخش های پروژه ، باید منطق پیاده سازی مشخصی داشته باشید. ارائه **داکیومنت** از بخش های پروژه شامل توصیف کد و خروجی الزامی می باشد.
- برای آشنایی با برنامه نویسی سوکت می توانید از منابع زیر استفاده 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

موفق و پیروز باشید.

مهدی صادقی نژاد ، بهاره حسینی ، مرتضی ملائی