

کتابخانه

فاز ۱

در این پروژه قصد داریم داستان یک کتابخانه را با استفاده از توانایی‌ها و مفاهیم سیستم عامل انجام دهیم. در این داستان، دو موجودیت اصلی نقش بازی می‌کنند. موجودیت اول یک کتابخانه است که خدماتی را ارائه می‌دهد و موجودیت دوم هم کاربران آن کتابخانه‌اند. در انتهای این پروژه باید یک سیستم (مجموعه‌ای از پردازه‌های در حال تعامل) ساخته شود تا اعمال کاربران و پاسخ‌های کتابخانه را به درستی شبیه‌سازی کند. جزئیات پیاده سازی در ادامه توضیح داده خواهد شد.

● سازنده

ساختار بالا به پایین سیستم به این صورت است که در هنگام شروع برنامه، فقط یک پردازه وجود دارد. این پردازه که آن را پردازه سازنده می‌نامیم، وظیفه دارد که فضا را برای شروع به کار بقیه پردازه‌ها آماده کند، تمام پردازه‌های جدید را بسازد و منتظر تمام شدن آن‌ها باشد.

یک فایل `log` هم باید درست شود که تمام پردازه‌ها اعمالی که در هر لحظه انجام می‌دهند را در آن یادداشت کنند. با توجه به این که تمام پردازه‌ها به این فایل دسترسی دارند، پیاده‌سازی آن باید طوری باشد که از دسترسی هم‌زمان به آن جلوگیری شود.

سازنده همچنین باید پل‌های ارتباطی اولیه که برای ارتباط کاربران با کتابخانه مورد استفاده قرار گرفته خواهد شد (مانند شماره پورت‌ها) را در اختیار آن‌ها قرار دهد.

● کتابخانه

این موجودیت نقش اصلی را در داستان بازی می‌کند و به کاربران خدمات می‌دهد.
کتابخانه ۵ نوع خدمت ارائه می‌دهد که در دو دسته قرار می‌گیرد:

1. خدمات مربوط به کاربر: ثبت‌نام کاربر / بررسی سطح دسترسی کاربر
2. خدمات مربوط به کتاب: اضافه کردن کتاب جدید / قرض دادن کتاب / پس گرفتن کتاب

پس از شروع به کار، برای هر کدام از این دو دسته، یک پردازه جدید ساخته می‌شود.
هر کدام لیستی مربوط به کار خود را نگه می‌دارند. این دو پردازه جدید نیز با استفاده از socket شبکه کار می‌کنند.

ارتباط کتابخانه با کاربران:

پردازه کتابخانه برای ارتباط با پردازه‌های کاربر از سوکت شبکه استفاده می‌کند که شماره پورت‌های آن، از قبل تعیین شده است. او منتظر برقرار شدن ارتباط توسط یکی از کاربران می‌شود. هر درخواست از طرف کاربران باید با استفاده از یک پردازه جدید پاسخ داده شود و کتابخانه باید باشد توانایی پاسخ به ۳ درخواست به طور همزمان را داشته باشد. دستورات ممکن به همراه مقدار برگشتی آنها در ادامه لیست شده است. هر کدام از دستورات می‌توانند خروجی موفق یا ناموفق تولید کنند.

register_user <user_name>

Returns:

success <user_id> •

failure •

register_book <book_title>

Returns:

success •

failure •

lend_book <user_id> <book_title>

Returns:

- success •
- failure •

return_book <user_id> <book_title>

Returns:

- success •
- failure •

۱. ثبت نام کاربر جدید:

در صورت موفقیت، کتابخانه توکن دسترسی کاربر را به لیست کاربران اضافه می‌کند و برای انجام خدمات، آن را درخواست خواهد کرد.

۲. احراز هویت:

انجام این کار باعث می‌شود کاربر بتواند به خدمات قرضدادن و پس‌دادن کتاب دسترسی داشته باشد.

۳. اضافه کردن یک کتاب جدید:

کاربران برای اضافه کردن کتاب نیازی به ثبت نام قبلی ندارند. بعد از این که کاربر کتاب را ارسال کرد، کتاب به لیست کتاب‌ها اضافه می‌شود.

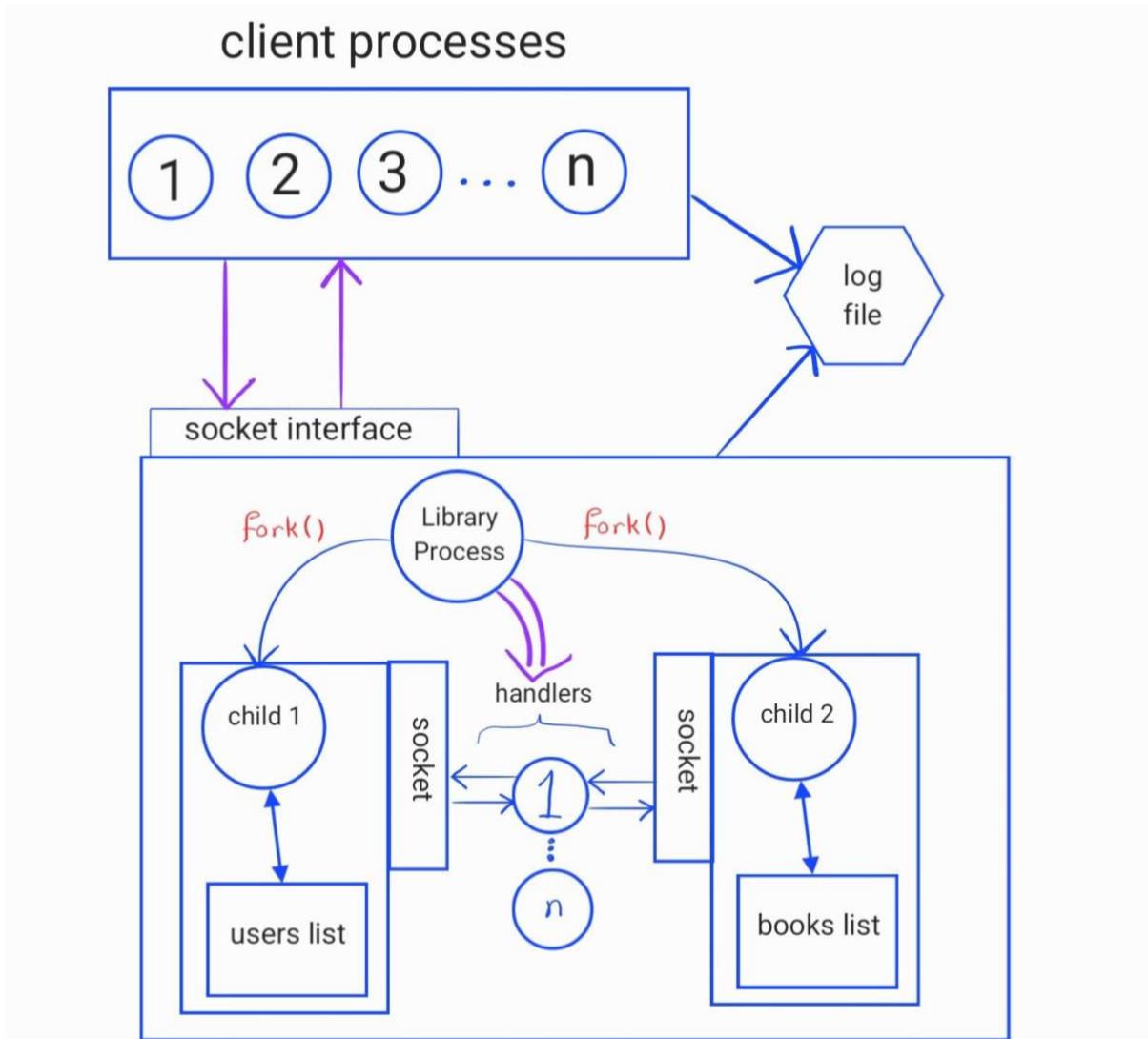
۴. قرض گرفتن کتاب:

ابتدا دسترسی کاربر چک می‌شود و در صورت وجود کتاب و در دسترس بودن آن، کتاب در اختیار این کاربر قرار خواهد گرفت.

۵. پس‌دادن کتاب:

دسترسی کاربر چک شده و دسترسی کتاب در لیست کتاب‌ها به روزرسانی می‌شود تا کاربران دیگر بتوانند آن را قرض کنند.

روند داخلی پردازه کتابخانه برای انجام خدمات:



• کاربر

موجودیت کاربر از خدمات کتابخانه استفاده می‌کند. برای شبیه سازی اعمال کاربر ها، برای هر کدام باید یک فایل دستورات از قبل آماده شده باشد که نشان دهد هر لحظه باید چه کاری انجام دهند. این فایل دستورات به این صورت است که در هر خط یک دستور وجود دارد. لیست دستورات قابل اجرا را ادامه آمده است.

Sleep <float-number (seconds)>

Register

title>_Register_book <book

title>_Lend <book
title>_Return <book

مثال: کاربری می‌خواهد ثبت‌نام کند، کتابی قرض بگیرد و بعد از مدتی آن را پس دهد.
فایل دستورات آن به صورت زیر خواهد بود:

Register
Lend Harry_Potter
Sleep 4.5
Return Harry_Potter

انتظار می‌رود بعد از پیاده‌سازی اولیه، سیستم را با استفاده از فایل‌های دستورات طولانی و پیچیده تست کنید تا از عملکرد صحیح آن، مخصوصاً وقتی چندین پردازه با هم درخواست ارسال می‌کنند، اطمینان حاصل شود.

نکات تكميلی:

- اين پروژه از نظر الگوريتمي بسيار ساده است با اين هدف که بر روی مفاهيم سیستم عامل و عملکرد توابعی که مستقيماً از قابلیت‌های سیستم عامل استفاده می‌کنند تمرکز و دقت بيشتری شود. از اين جهت انتظار می‌رود که رفتار توابع اين‌چنینی که استفاده می‌کنيد را به دقت تحقيق کنيد و از کارکرد دقیق آن‌ها اطلاع حاصل کنيد.
- زبان پیشنهادی برای این پروژه python است. می‌توانید از هر زبان دیگری که اجازه فراخوانی توابع لازم برای استفاده از قابلیت‌های سیستم عامل را می‌دهد نیز استفاده کنید از جمله go, C/C++, rust, ...
- زبان‌هایی مانند javascript برای این پروژه قابل استفاده نمی‌باشند. در صورت استفاده از زبان نامناسب، نمره ای به پروژه داده نمی‌شود.
- برای انجام این پروژه باید از یک سیستم‌عامل UNIX-like استفاده کنید. در صورتی که می‌خواهید از ویندوز استفاده کنید باید با ابزار WSL کار کنید.