

سرویس اطلاع رسانی ساده

از قابلیت های مهم برنامه نویسی سوکت قابلیت بی درنگ یا **real time** بودن آن است. برای مثال فرض کنید یک برنامه برای گفتگو آنلاین میخواهیم پیاده سازی کنیم. درحالت عادی برای اینکه کاربران از پیامهایی که بهشون ارسال شده مطلع بشن باید مدام به سرور درخواست ارسال کنند(مثلا هر 5 ثانیه)!

حالا فرض کنید برنامه ای که نوشتیم 10000 کاربر آنلاین داشته باشد و هر کاربر هر 5 ثانیه درخواست گرفتن آخرین پیام ها و تغییرات را به سرور ارسال کند! بیشتر اوقات درصد پایینی از این 1000 کاربر تغییرات جدیدی دارند که باید از سرور دریافت کنند و بقیه کاربرها فقط بعنوان بار اضافی روی سرور محسوب میشوند.

برای حل این مشکل میشود از سوکت پروگرامینگ استفاده کرد! به این صورت که یک **server socket** میسازیم که کلاینت ها بتوانند به آن متصل شوند و سرور تغییرات هر کاربر را (بدون اینکه کلاینت نیازی به ارسال درخواست داشته باشد) به آن اطلاع میدهد.

از مزیت های این روش:

1. **Overhead** بسیار کمتر

2. **realtime** بودن. یعنی تغییرات در لحظه به کاربر اعلام میشود بدون اینکه خود کاربر درخواستی به سرور ارسال کند.

البته این روش معایبی هم نظیر افت سرعت درصورت بالا رفتن بیش از حد تعداد کاربران متصل به سوکت دارد.

پروژه:

هدف این پروژه ساخت یک سرویس **push notification** ساده توسط سوکت پروگرامینگ است.

تعریف Push notification:

در بعضی پروژه های نرم افزاری ممکن است نیاز داشته باشیم تا سرور ، کلاینت را از آخرین تغییراتی که مرتبط با آن است با خبر کند. برای مثال فرض کنید ما یک سیستم فروشگاهی داریم و می خواهیم به محض اینکه محصولی در فروشگاه اضافه شد به بعضی از کاربران سیستم اعلام کنیم که محصولی مرتبط با علائق شما به سیستم اضافه شده است. منبع

در این تمرین قسمت مربوط به اتصال پایگاه داده خواندن و نوشتن آن برای شما آماده شده است. کاری که قرار است انجام دهید این است که یک سرور چند نخی multi-threaded server بسازید.

کار سرور به این صورت باید باشد که هنگامی یک کلاینت درخواست اتصال را ارسال میکند برایش یک نخ جدید ساخته و هرگاه اپدیت جدیدی مربوط با userID آن وجود داشت کاربر را مطلع کند.

بنابراین کارهایی که برای تکمیل پروژه انجام خواهید داد به شرح زیر است:

بخش سرور: (فایل server.py را تکمیل کنید)

1. یک socket server ساخته و در یک حلقه بی پایان منتظر client جدید باشید
2. هنگامی که کلاینت متصل شد یک thread جدید ساخته و آن را به کلاینت اختصاص دهید و دوباره منتظر کاربر جدید شوید
3. یک thread بسازید که هر 5 ثانیه با استفاده از تابع check_database_for_new_push_messages تغییرات اخیر دیتابیس را دریافت کند و طبق آن به کلاینت های مربوطه پیام جدید را اطلاع دهد.
4. پس از انجام مرحله قبل در صورتی که کاربر مربوطه به سرور متصل بود تابع `update_visited_notification(username, message_id)` را صدا بزنید

بخش کلاینت (فایل client.py را تکمیل کنید)

1. یک socket بسازید که به سرور متصل شود
2. منتظر ماندن برای دریافت پیام جدید
3. بستن اتصال پس از 1 دقیقه

نکات قابل توجه:

- 😊 این پروژه بعنوان نمره مثبت در نظر گرفته شده است و در صورت انجام ندادن آن نمره‌ای کسر نمیشود.
- 😊 هر بخش از این پروژه بین 5 تا 20 نمره دارد و نمره کل از 100 محاسبه میشود.
- 😊 آخرین زمان ارسال پروژه ساعت 23:59 شنبه 8 دی ماه میباشد.
- 😊 برای ارسال پروژه میبایست در کلاس مجازی quera.ir ثبت نام کنید*.
- 😊 فقط فایل‌های server.py و client.py را تغییر دهید و برای اضافه کردن داده جدید به دیتابیس database_interface.py را اجرا کنید.
- 😊 برای دریافت فایل پروژه به آدرس <https://github.com/mehran75/SocketProgramming-presentation/tree/master/assignment> مراجعه کنید.

😞 کدها بصورت نرم افزاری مورد آزمایش قرار گرفته و هرگونه شباهت بین آنها مشخص خواهد شد.

درس شبکه های کامپیوتری پیشرفته - دکتر شکران

طراح سوال: مهراڢ رفیعۃ

