

دلال پیام - قسمت اول

پرهام الوانی

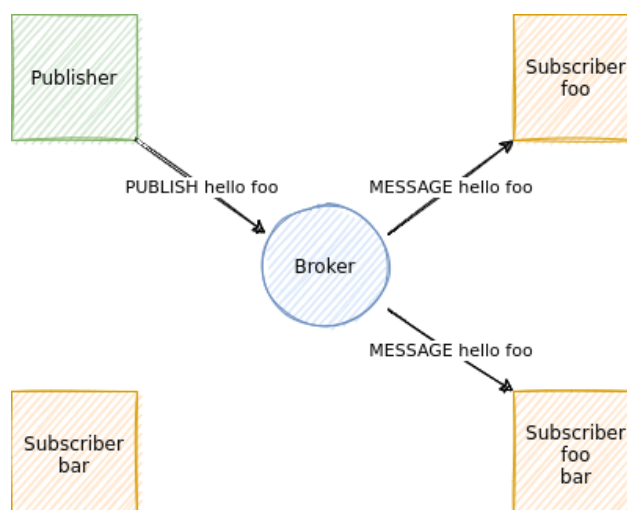
۲ آذر ۱۴۰۰

فهرست مطالب

۲	۱ مقدمه
۲	۲ دستورات
۳	۳ پیاده‌سازی سرور
۳	۴ پیاده‌سازی کلاینت
۴	۵ امتیازها
۴	۶ تحویل‌دادنی‌ها
۴	۷ نکات

۱ مقدمه

دلال پیام^۱ یک سرور است که پیام‌های دریافتی از کلاینت‌ها را بین آن‌ها تقسیم می‌کند. البته این دلال‌ها کارهای بیشتری هم انجام می‌دهند که خارج از بحث ما است. دو عملیات مهم در سرور وجود دارد یکی Publish کردن و دیگری Subscribe کردن است. در عملیات Subscribe کلاینت درخواست گوش دادن رو یک موضوع^۲ خاص را می‌دهد. در Publish کلاینت یک پیام را روی یک موضوع خاص منتشر می‌کند. در نهایت دلال پیام‌ها را به دست تمام کسانی که روی آن موضوع گوش می‌دهند می‌رساند. در این قسمت تمامی ارتباط میان سرور و کلاینت در قالب TCP پیاده‌سازی می‌شود.



شکل ۱: شمای کلی یک دلال پیام

۲ دستورات

کلاینت به سرور:

Publish

این دستور یک پیام از سمت کلاینت را تحت یک عنوان خاص برای سرور ارسال می‌کند. پیام‌ها رشته‌هایی ASCII با طول دلخواه هستند.

Subscribe

این دستور به سرور اعلام می‌کند که این کلاینت متقاضی دریافت پیام‌های، عنوان داده شده است.

Ping

این دستور از سمت کلاینت برای اطمینان از ارتباط ارسال می‌گردد.

Pong

این دستور از سمت کلاینت در پاسخ به پیام Ping ارسال می‌گردد.

سرور به کلاینت:

Message

این دستور یک پیام از سمت سرور را برای کلاینت زمانی که متقاضی موضوعی است، ارسال می‌کند (دقت داشته باشید که این دستور شامل پیام و عنوان است). پیام‌ها رشته‌هایی ASCII با طول دلخواه هستند.

SubAck

این دستور از سمت سرور در جهت تایید پیام Subscribe کلاینت زمانی که عملیات Subscribe موفقیت آمیز باشد ارسال می‌شود.

PubAck

این دستور از سمت سرور در جهت تایید پیام Publish کلاینت زمانی عملیات Publish موفقیت آمیز باشد ارسال می‌شود.

¹Message Broker

²Topic

Ping

این دستور از سمت سرور برای اطمینان از ارتباط ارسال می‌گردد.

Pong

این دستور از سمت سرور در پاسخ به پیام Ping ارسال می‌گردد.

۳ پیاده‌سازی سرور

سرور یک سوکت سرور دارد که روی یک پورت مشخص گوش می‌دهد (در اینجا فرض می‌کنیم پورت موردنظر ۱۳۷۳ است). از آنجایی که قصد داریم سرور چند ارتباط همزمان را هندل کند نیاز به ساخت Thread برای هر کانکشن خواهید داشت تا تابع accept بلافاصله فراخوانی شود. سرور هیچ ارتباطی را نمی‌بندد و همه ارتباط‌ها را به صورت باز نگه می‌دارد. برای سرور هر ارتباط یک کلاینت را نمایندگی می‌کند بنابراین می‌بایست لیست از سوکت‌ها و عناوین موردنظر آن‌ها داشته باشد.

```
import socket
import threading

HOST = '127.0.0.1' # Standard loopback interface address (localhost)
PORT = 1373 # Port to listen on (non-privileged ports are > 1023)

def handler(conn, addr):
    with conn:
        print('Connected by', addr)
        while True:
            data = conn.recv(1024)
            if not data:
                break
            conn.sendall(data)
        print('Disconnected by', addr)

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    while True:
        conn, addr = s.accept()
        threading.Thread(target=handler, args=(conn, addr)).start()
```

دقت داشته باشید که سرور می‌بایست وضعیت کانکشن‌ها را نیز مدنظر داشته باشد و از ارسال داده بر روی کانکشن‌های بسته شده خودداری کند.

۴ پیاده‌سازی کلاینت

کلاینت به صورت یک ابزار درون ترمینالی پیاده‌سازی می‌شود و نیازی به پیاده‌سازی گرافیکی **نیست**. کلاینت آرگومان‌های زیر را در زمان اجرا دریافت می‌کند. برای اطلاعات بیشتر از آرگومان‌ها در پایتون می‌توانید از اینجا استفاده کنید. کلاینت دو دستور اصلی دارد:

دستور Publish پیام داده شده در آرگومان message را تحت عنوان آرگومان topic به سرور ارسال می‌کند. این دستور به مدت **۱۰ ثانیه** تا رسیدن پیام PubAck از سمت سرور صبر می‌کند و در صورت دریافت موفقیت آمیز، به کاربر اطلاع می‌دهد. در غیر این صورت خطای مناسبی نمایش داده می‌شود.

```
> python3 client.py <host> <port> publish <topic> <message>

> your message published successfully # in case of success

> your message publishing failed # in case of failure
```

دستور Subscribe به سرور اطلاع می‌دهد که متقاضی دریافت پیام‌های عنوان‌های داده شده تحت آرگومان‌های topic هستیم. دقت کنید این دستور را می‌شود با یک یا بیشتر topic فراخوانی کرد و شما نیاز دارید برای داشتن همزمان چند کانکشن با سرور از Thread (برای اطلاعات بیشتر می‌توانید به اینجا مراجعه کنید) استفاده کنید. برای هر عنوان شما نیاز به یک عملیات Subscribe شما نیاز به ارسال یک پیام Subscribe و دریافت یک پیام SubAck در مدت محدود **۱۰ ثانیه** دارید. در صورتی که پیام SubAck در این مدت محدود دریافت نشود برنامه شما می‌بایست با پیام مناسب خاتمه پیدا کند. در نهایت برنامه می‌بایست روی همه سوکت‌های ساخته شده گوش فرا دهد تا پیام‌های دریافتی در قالب دستور Message را به کاربر نمایش دهد.

```
> python3 client.py <host> <port> subscribe <topic-1> <topic-2> ... <topic-n>

> # in case of success
> subscribing on <topic-1> <topic-2> ... <topic-n>
> <topic-1>: message
> <topic-2>: message
> <topic-1>: message

> # in case of failure
> subscribing failed
```

۵ امتیازها

- کلاینت در هر دو حالت می‌بایست به پیام‌های Ping از سمت سرور با پیام Pong پاسخ دهد.
- سرور می‌بایست به صورت دوره‌های **۱۰ ثانیه‌ای** پیام‌های Ping به کلاینت ارسال کند و در صورت نگرفتن جواب تا شروع دوره‌ی بعدی، اطلاعات آدرس آی پی و پورت کلاینت مورد نظر را چاپ کند.
- در صورتی که کلاینت سه دوره متوالی به پیام Ping پاسخ نداد کانکشن کلاینت بسته شده و از لیست کلاینت‌ها حذف شود.

۶ تحویل‌دادنی‌ها

- طراحی و توضیح پروتکل و نحوه ارسال داده‌ها. در این تمرین عملی دستورات متعددی تعریف شده است و شما می‌بایست مشخص کنید که این دستورات را چگونه بر روی ارتباط TCP بین کلاینت و سرور جابجا می‌کنید.
- پیاده‌سازی سرور با زبان برنامه‌نویسی مورد نظر شما
- پیاده‌سازی کلاینت با زبان برنامه‌نویسی مورد نظر شما

۷ نکات

- تمرین‌های عملی را می‌بایست به صورت **انفرادی** انجام دهید.
- تمرین عملی نباید هیچ‌گونه exception یا خطای ناخواسته‌ای داشته باشد. تمام خطاهای شما می‌بایست به درستی هندل شده یا حداقل در کد دارای کامنت باشند و علت وقوع آن ذکر شده باشد.
- هرگونه کپی‌برداری یا تقلب در صورت تشخیص نمره طرفین را صفر خواهد کرد.
- آموزش برنامه‌نویسی سوکت با پایتون در آزمایشگاه انجام شده است و لینک‌های لازم برای این زبان نیز در این صورت تمرین وجود دارد ولی انتخاب زبان برنامه‌نویسی آزاد است.

این سند برپایه بسته X3Persian گونه 23.1 توسعه پیدا کرده است.