FTP Server

محمد عراقی 810198436

محمد سعادتی 810198410

مقدمه

هدف از این پروژه آشنایی با مفاهیم socket programming و ارتباط از طریق socket ها بوده

است . کلاینت ها با دانستن پورت سرور به سرور وصل شده و ارتباط برقرار میکنند و بسته به نوع command ها ، اطلاعات و پیام های متناسب میان سرور و هر کلاینت رد و بدل میشود .

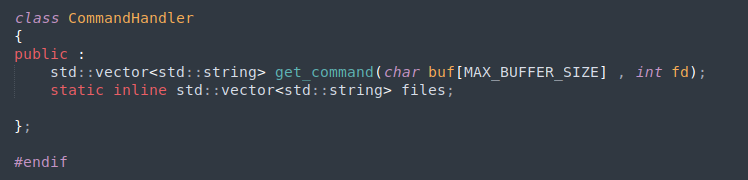
کلاس ها و متد ها

1) CommandHandler

وظیفه ی این کلاس تشخیص دستور ارسال شده برای سرور و پاسخ مناسب به آن است و بخش بیشتر پروژه به عهده ی آن است .

این کلاس دارای وکتوری ازنام فایل ها (فایل های مجاز کاربران ) و همچنین تابع get command است .

تابع get command با هربار دریافت داده توسط سرور صدا زده میشود و پیام ارسالی را تحلیل میکند و پاسخ مناسب را آماده ی ارسال میکند . ورودی این تابع ، بافر سرور (پیام ارسال شده توسط کلاینت) و file descriptor کلاینت فرستنده ی پیام است.



2) ConfigurationParser

اطلاعات مربوط به فایل config در این کلاس نگه داری شده و در صورت نیاز با دسترسی به آن میتوان به اطلاعاتی اعم از اطلاعات یوزرها ، فایل های مجاز ، و همچنین پورت های چنل دیتا و کامند دسترسی یافت .

Attributes:

command\_channel\_port: پورت چنل کامند (مورد نیاز برای کلاینت ها)

data\_channel\_port: پورت چنل دیتا (مورد نیاز برای کلاینت ها)

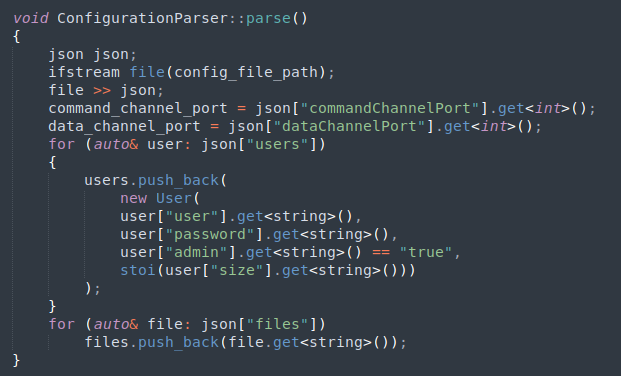
config\_file\_path: مسیر فایل config.json

users: وکتوری از یوزرهایمان که اطلاعات خوانده شده ی یوزر ها در آن نگه داری میشود

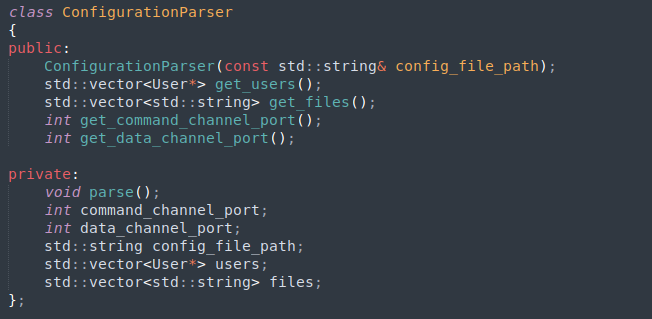
files: وکتوری از نام فایل های مجاز یوزرهایمان

Methods:

Parse: برای خئاندن اطلاعات فایل config.json صدا زده میشود و به شکل زیر عمل میکند :



که همانگونه که مشخص است اطلاعات را خوانده و به ازای هر یوزر یک یوزر ساخته و به وکتور یوزرها اضافه میکند .



3) ConnectedUser

اطلاعات مربوط به یوزر هایی که به سرور متصل شده اند را نگه میدارد ، اعم از وضعیت login ، directory کنونی آن و ...

Attributes:

User: یوزر مربوطه در وکتور users در ConfigurationParser

fd: file descriptor یوزر وصل شده به سرور

command\_channel\_socket\_fd و data\_channel\_socket\_fd:

fd های کانال دیتا و کامند

is\_user\_name\_entered و is\_password\_entered:

بزای بررسی وضعیت لاگین و اینکه در کدام مرحله ی لاگین است . (username و password وارد شده اند یا خیر).

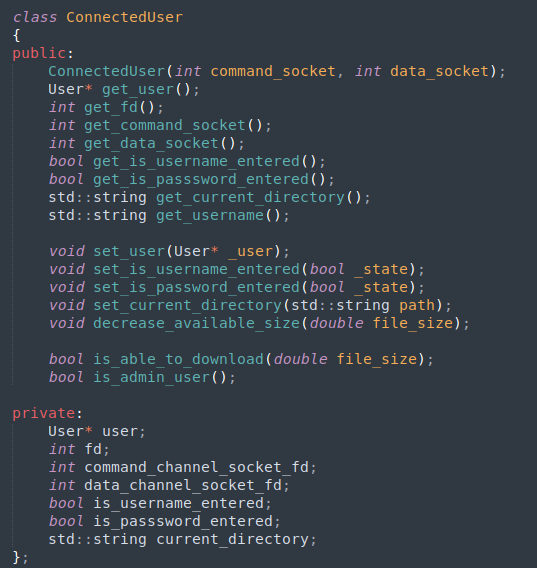
current\_directory: directory کنونی کلاینت مان

Methods:

decrease\_available\_size: برای کاهش حجم کنونی یوزر پس از دانلود فایل

is\_able\_to\_download: برای بررسی ملزومات کافی برای دانلود فایل

is\_admin\_user: برای بررسی admin بودن یا یوزر عادی بودن



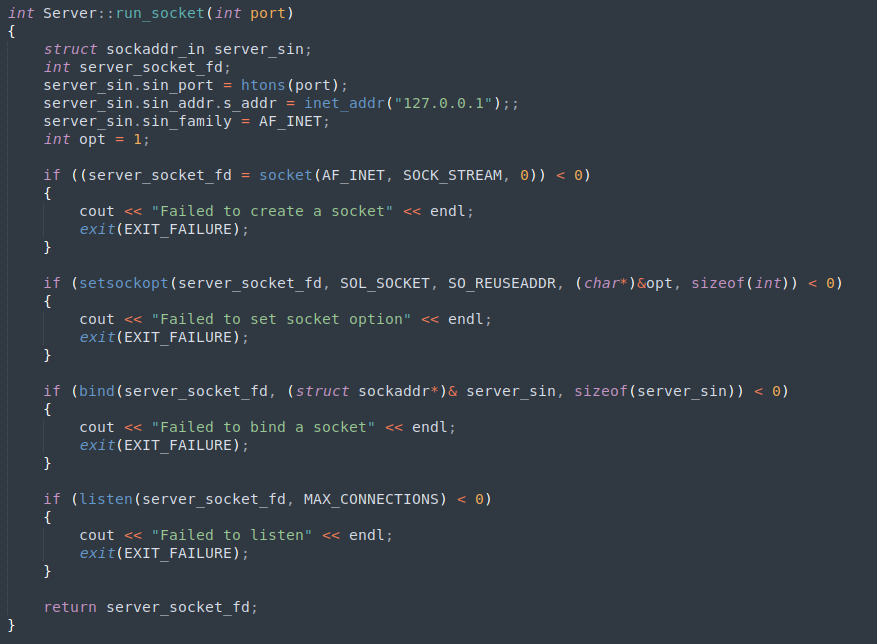
4) Server

برای ایجاد و بایند کردن سرورمان.

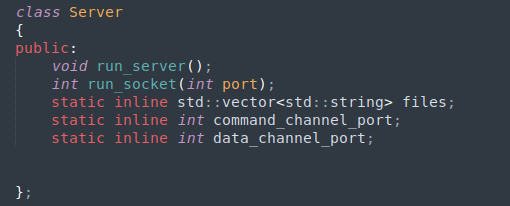
Methods:

run\_server: برای برقراری سرور

run\_socket: برای بایند کردن سرور به سوکت مناسب و بقیه کارهای اجرایی که به شکل زیر است:



و طبق گفته های کلاس پیاده سازی شده است .



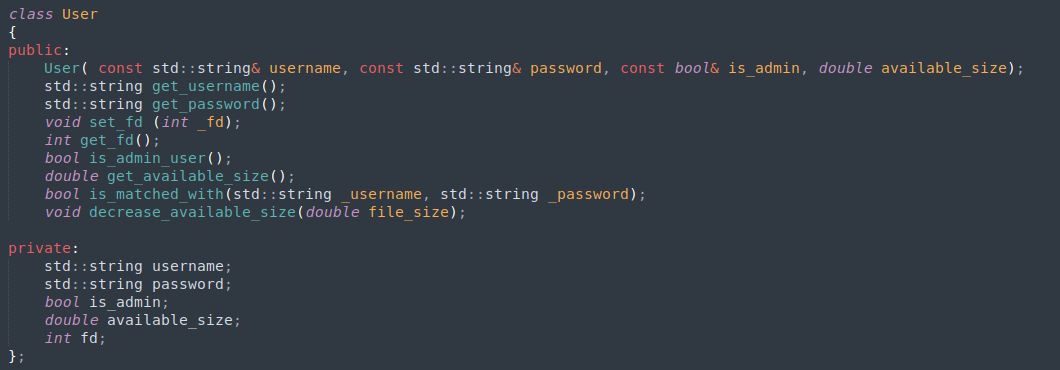
5) UserManager

تمام کارهای مربوط به مدیریت یوزرها ، اعم از اطلاعات اولیه یوزرها و وضعیت کنونی آن ها ، پیدا کردن یوزر ها بر اساس پارامتر های مناسب ، حذف و اضافه یوزرها و ... به عهده این کلاس است و به users و connected\_users دسترسی دارد .



6) User

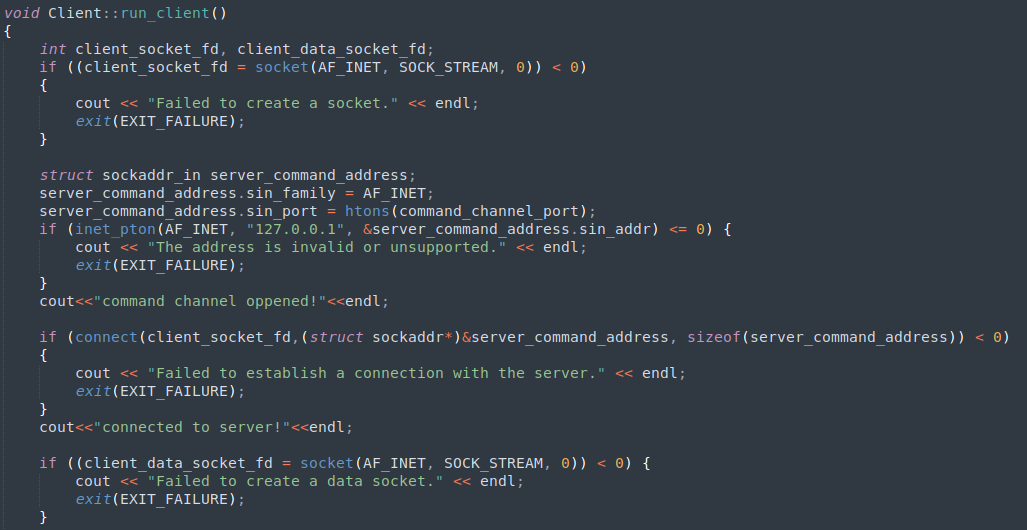
همانند connected\_user ، با این تفاوت که اطلاعات اولیه یوزرها را که از فایل کانفیگ خوانده شده شامل است و به ئضعیت کنونی یوزرها ارتباطی ندارد.



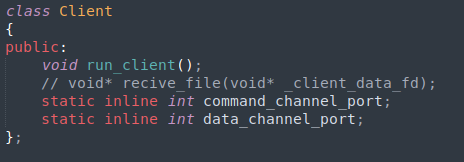
7) Client

کلاس کلاینت مان که اجرا شده و با داشتن پورت های لازم به سرور متصل شده و شروع به رد و بدل اطلاعات میکند .

run\_client: برای اجرای کلاینت و ساختن کلاینت جدید به شکل زیر:



که به شکل گفته شده در کلاس پیاده سازی شده است .



دیگر توابع :

exec: برای اجرای کامند ها در سطح کرنل به وسیله ی پایپ ، و بازگرداندن نتیجه ی به دست آمده به عنوان result

