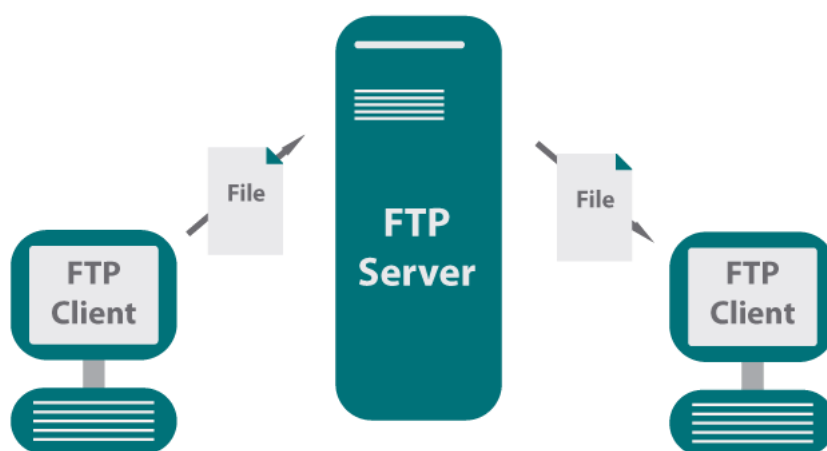




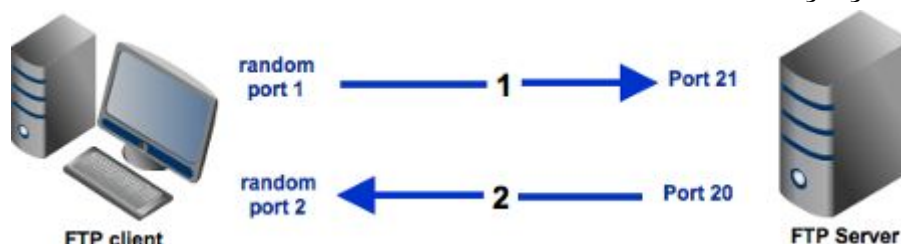
FTP سرور



در این پروژه، شما به پیاده‌سازی یک برنامه در لایه‌ی کاربردی یعنی یک ورژن ساده‌شده‌ی ftp می‌پردازید. این برنامه از دو بخش client و server تشکیل شده‌است که server وظیفه‌ی خدمت‌رسانی فایل‌ها را به client دارد.

این پروتکل چگونه کار می‌کند؟

ارتباط ftp به دو کانال مختلف برای ارسال و دریافت داده‌ها نیاز دارد. کانال اول که دستور^۱ نامیده می‌شود کانالی است که دستورها و پاسخ آن‌ها از آن رد می‌شود. کانال دیگر که کانال داده^۲ نامیده می‌شود، کانالیست که وظیفه‌ی جابه‌جایی داده را دارد.



^۱ command channel

^۲ data channel

مقدمه

شما باید در این پروژه یک برنامه‌ی سرور با دو کانال ارتباطی ایجاد کنید. این دو کانال ارتباطی می‌توانند روی دو پورت دلخواه که از طریق فایل کانفیگ به برنامه‌ی سرور داده می‌شوند اجرا شوند. سپس شما به پیاده‌سازی یک کلاینت که از طریق دو پورت دلخواه به این کانال‌ها متصل می‌شود می‌پردازید. توجه داشته باشید که سرور وظیفه سرو کردن فایل‌های کامپیوتری که در آن در حال اجراست را دارد. همین‌طور محلی که برنامه قرار دارد و اجرا می‌شود به عنوان دایرکتوری اولیه در نظر گرفته می‌شود.

احراز هویت³

برای احراز هویت در سرور باید از فایل json مربوط به کانفیگ سرور اطلاعات کاربران سیستم را خوانده و با استفاده از پسورد آن‌ها را شناسایی و احراز هویت کنند.

```
{
  "users": [
    {
      "user": "navid",
      "password": "1234"
    }
  ]
}
```

در کلاینت برای اینکه یک کاربر وارد شود باید ابتدا نام کاربری خود را مشابه دستور زیر وارد کند:

USER <username>

در صورتی که username قابل قبول بود پاسخ زیر داده می‌شود.

331 User name okay, need password.

سپس سرور منتظر رمز عبور کاربر می‌ماند. در این حالت کلاینت باید رمز عبور را برای سرور ارسال نماید.

PASS <password>

در صورتی که کلاینت قبلاً درخواست ورود نداده بود پاسخ زیر داده می‌شود.

³ Authentication

503 Bad sequence of commands.

در صورتی که کاربر با موفقیت وارد می شد پاسخ زیر داده می شود.

230 User logged in, proceed.

در صورتی که هر کدام از رمز عبور یا نام کاربری ایراد داشتند و قابل قبول نبودند پاسخ زیر داده می شود.

430 Invalid username or password.

دستورات

بعد از اینکه کاربر وارد شد می تواند با ftp سرور کار کند و دستوراتی که در ادامه آمده است را انجام دهد.

دایرکتوری فعلی (PWD)

این دستور دایرکتوری فعلی که در آن هستید را به شما نمایش می دهد:

PWD

پاسخ این دستور در کانال دستور به شکل زیر ارسال می شود.

257 <working directory path>

ساختن فایل یا دایرکتوری جدید

این دستور یک فایل یا دایرکتوری جدید با نام گرفته شده در سرور ایجاد می کند. در صورتی که از پرچم ⁴ -i استفاده شده باشد یک فایل جدید و در غیر این صورت یک دایرکتوری جدید ساخته می شود.

MKD -i <name>

پاسخ این دستور در کانال دستور به شکل زیر ارسال می شود.

257 <filename/directory path> created.

پاک کردن فایل یا دایرکتوری

این دستور یک فایل یا دایرکتوری موجود را حذف می کند. در صورتی که از پرچم -f استفاده شده باشد یک دایرکتوری را حذف می کند و در غیر این صورت فایل را حذف می کند.

⁴ Flag

RMD -f <name>

پاسخ این دستور در کانال دستور به شکل زیر ارسال می شود.

250 <filename/directory path> deleted.

لیست فایل های موجود در دایرکتوری

با اجرای این دستور کاربر لیست فایل های موجود در دایرکتوری کنونی را مشاهده می کند.

LIST

لیست فایل ها از طریق **کانال داده** برای کلاینت ارسال می شود. همین طور پس از ارسال کامل لیست فایل ها در کانال دستور پاسخ زیر ارسال می شود. در کلاینت پس از نمایش پاسخ باید لیست فایل های موجود را نمایش دهید.

226 List transfer done.

عوض کردن دایرکتوری

با اجرا کردن این دستور می توانید بین دایرکتوری های موجود جابه جا شوید. دقت داشته باشید در صورتی که پس از دستور "..." آمد به این معنی است که باید به دایرکتوری قبلی بروید. همچنین در صورتی که این دستور بدون آرگومان وارد شد باید به دایرکتوری اولیه بروید.

CWD <path>

پاسخ این دستور در کانال دستور به شکل زیر ارسال می شود.

250 Successful Change.

دانلود فایل

این دستور فایل گفته شده را در صورتی که موجود باشد دانلود می کند.

DL <name>

فایل از طریق **کانال داده** برای کلاینت ارسال می شود و پس از آن که انتقال فایل کامل شد پاسخ زیر از طریق کانال دستور ارسال می شود.

226 Successful Download.

راهنما

این دستور، دستورات موجود در سرور را به همراه راهنمای استفاده از آن‌ها به کاربر نمایش می‌دهد.

HELP

پاسخ این دستور در کانال دستور ارسال می‌شود و به عنوان مثال مانند شکل زیر است. شما باید همه دستورات را مانند دستور زیر در پاسخ این دستور ارسال کنید.

214

USER [name], Its argument is used to specify the user's string. It is used for user authentication.

خارج شدن از سرور

این دستور کاربر فعلی را از سیستم خارج می‌نماید.

QUIT

پاسخ این دستور در کانال دستور به شکل زیر ارسال می‌شود.

221 Successful Quit.

مدیریت خطاها

در تمامی حالات اگر کاربری هنوز وارد نشده بود و دستورات را وارد کرد، سرور باید پاسخ زیر را برگرداند.

332 Need account for login.

در تمامی حالات اگر ایراد نگارشی در پارامترهای یک دستور وجود داشت باید پاسخ زیر داده شود.

501 Syntax error in parameters or arguments.

در صورتی که خطای دیگری رخ داد پاسخ زیر را در کانال دستور ارسال کنید.

500 Error.

همین‌طور اگر علاقه‌مند بودید می‌توانید با استفاده از [این](#) لینک تمامی کدهای پاسخ را مشاهده کنید و در هر مورد از خطاها پاسخ مناسب به آن را برگردانید.

کلاینت

شما باید یک کلاینت (می‌تواند یک برنامه تحت خط فرمان⁵ باشد) پیاده‌سازی کنید تا با استفاده از دستورات گفته شده با سرورتان بتواند ارتباط برقرار کند و این دستورات را انجام دهد. توجه کنید هر پیاده‌سازی معقولی قابل قبول می‌باشد.

همین‌طور توجه داشته باشید برای راحتی کار پاسخ‌هایی که از طریق کانال داده ارسال می‌شود می‌تواند به صورت قراردادی بین سرور و کلاینت از هر فرمتی پیروی کند.

امکانات ویژه

حسابداری (Accounting)

در فایل کانفیگ سرور می‌توان این امکان را روشن کرد. در صورتی که این امکان روشن باشد برای یکی سری از کاربران می‌توان حجم مصرفی در نظر گرفت و در صورت اتمام به کاربران ایمیل ارسال می‌شود. این امکان در فایل کانفیگ به صورت زیر مشخص می‌شود.

```
{
  "accounting": {
    "enable": true,
    "threshold": 100,
    "users": [
      {
        "user": "farzad",
        "size": "12000",
        "email": "farzad@habibi.com",
        "alert": true
      }
    ]
  }
}
```

در بخش users کاربران به همراه حجمی که می‌توانند مصرف کنند در واحد Byte داده می‌شود. در هنگام ورود دستور دانلود یک فایل، در صورتی که کاربر حجمی کافی برای دریافت آن فایل داشت، فایل برایش ارسال می‌شود و مقدار حجم فایل از حجم او کم می‌شود. در صورتی که حجم کاربر کافی نبود پاسخ زیر از سمت سرور داده می‌شود:

⁵ command line

425 Can't open data connection.

ارسال ایمیل به کاربر

اگر سرویس alert برای کاربر فعال باشد در صورتی که حجم یک کاربر کمتر از threshold شد، باید به کاربر ایمیل اتمام حجم ارسال شود. این ایمیل باید از طریق پروتکل SMTP به ایمیلی که در فایل کانفیگ مشخص شده است ارسال شود.

نحوه‌ی ارسال ایمیل

برای ارسال ایمیل می‌توانید از سرور mail دانشگاه استفاده کنید. قبل از شروع می‌توانید با استفاده از telnet اقدام به ارسال ایمیل کنید، بعد از مطمئن شدن از درستی کار می‌توانید با استفاده از socket در زبان برنامه‌نویسی خود اقدام به ارسال ایمیل کنید.

برای پیدا کردن server mail می‌توانید دستورات زیر را در ترمینال وارد کنید:

```
$ nslookup
> set type=mx
> <domain (e.g ut.ac.ir)>
```

برای متصل شدن به سرور mail می‌توانید با دستور 25 mail.ut.ac.ir telnet در ترمینال اقدام به شروع کردن مکالمه با سرور کنید. برای معرفی خود از دستور زیر استفاده کنید:

```
HELO <name>
```

برای وارد کردن آدرس فرستنده

```
MAIL FROM: <"sender email address">
```

در این قسمت لازم است برای احراز هویت نام کاربری و رمز خود را وارد کنید:
با دستور

```
AUTH LOGIN
```

برای احراز هویت اقدام کنید. باید دقت داشته باشید که نام کاربری باید به فرمت Base64 باشد. باید توجه داشته باشید که نام کاربری همراه با کاراکتر (\n) را به Base64 تبدیل کنید. مثلاً:

```
admin\n : YWRtaW4K
```

برای تبدیل رمز به فرمت Base64 تنها خود رمز را به Base64 تبدیل کنید. مثلاً:

```
admin: QWRtaW4=
```

در این مرحله باید آدرس گیرنده را وارد کنید:

```
RCPT TO: <Recipient email address>
```

اگر با پیام تایید مواجه شدید می‌توانید داده با زدن دستور زیر متن خود را برای سرور ایمیل کنید:

```
DATA
```

اگر با مشکلی روبرو شدید با جستجویی ساده در اینترنت می‌توانید آموزش‌های کامل‌تری بیابید.

در این بخش حق استفاده از کتابخانه‌های آماده مانند Smtplib و ... را ندارید و باید به صورت خام ایمیل را ارسال کنید.

Logging

سرور شما در زمان اجرا باید یک فایل log در کنار خود ایجاد کند و تمامی اطلاعات را با تاریخ و ساعت وقوع در آن ذخیره کند. اطلاعاتی مانند افرادی که وارد سیستم شده‌اند، فایل‌هایی که ساخته‌اند یا پاک کرده‌اند و فایل‌هایی که دانلود کرده‌اند.

ابتدا از فایل کانفیگ وضعیت این سرویس را بررسی کنید و در صورتی که سرویس فعال باشد باید در فایلی که مشخص شده است لاگ‌های خود را بنویسید. (اگر فایل وجود ندارد آن را بسازید و در صورتی که از قبل این فایل وجود داشته است در ادامه‌ی آن شروع به نوشتن کنید)

امتیازی - Authorization

در این بخش قصد داریم یک مکانیزم مدیریت دسترسی پیاده‌سازی کنیم. دو نوع کاربر در سیستم وجود دارند: ۱- ادمین ۲- کاربر عادی. کاربران عادی به فایل‌های عادی دسترسی دارند و ادمین به همه فایل‌های سیستم دسترسی دارد. به عنوان مثال تنها ادمین به فایل با آدرس `/x.png` می‌تواند دسترسی داشته باشد. در فایل `config.json` لیست فایل‌هایی که فقط ادمین سیستم به آن‌ها دسترسی دارد آمده است. همین‌طور در این فایل لیست کاربرانی که نقش ادمین را در سیستم دارند آمده‌است. در صورتی که کاربر به فایلی دسترسی نداشته باشد و بخواهد از آن‌ها استفاده کند باید پیام زیر را به او نمایش دهید.

550 File unavailable.

نکات پایانی

- پروژه در گروه‌های دو نفره انجام می‌شود.
- برنامه شما باید درخواست‌ها را به صورت هم‌زمان پاسخ دهد و چندین کلاینت بتوانند با کاربران مختلف به سرور وصل شوند. (می‌توانید از `thread` استفاده کنید)
- از هر زبانی می‌توانید استفاده کنید ولی کدهای شما باید سطح پایین و با استفاده از `socket` نوشته شوند.
- ساختار صحیح و تمیزی کد برنامه بخشی از نمره این پروژه شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.
- استفاده از زبان‌هایی مانند `c++` تنها به پیچیدگی کدهای شما می‌افزاید که از هدف پروژه به دور است، اکیدا پیشنهاد می‌شود از زبان `python` استفاده کنید.
- سؤالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن‌ها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این پروژه ارتباط برقرار کنید. توجه

داشته باشید که دیگر شبکه‌های اجتماعی مانند تلگرام راه ارتباطی رسمی با دستیاران آموزشی نیست و دستیاران آموزشی موظف به پاسخگویی در محیط‌های غیررسمی نیستند.

• ایمیل طراحان پروژه:

• anavid.akbari@gmail.com

• farzadhabibii98@gmail.com