



FTP سرور

در این پروژه شما به پیاده‌سازی 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 دو کانال مختلف برای ارسال و دریافت داده‌ها نیاز دارد. کانال اول که کانال دستور نامیده می‌شود کانالی است که دستورها و پاسخ آن‌ها از آن رد می‌شود. کانال دیگر که کانال داده نامیده می‌شود، کانالیست که وظیفه‌ی جابه‌جایی داده را دارد.

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 هویت^۱ و مدیریت دسترسی^۲

^۱ Authentication

^۲ Authorization

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

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

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

user <username>

در صورتی که username قابل قبول بود پاسخ زیر داده می شود.

331: User name okay, need password.

سپس سرور منتظر رمز عبور کاربر می ماند. 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 می شود.

503: Bad sequence of commands.

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

230: User logged in, proceed. Logged out if appropriate.

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

در ادامه‌ی این بخش قصد داریم یک مکانیزم مدیریت دسترسی پیاده‌سازی کنیم به طوری که دو نوع کاربر در سیستم وجود داشته باشند: ۱- ادمین ۲- کاربر عادی. کاربران عادی تنها به برخی فایل‌ها دسترسی دارند و این درحالی است که ادمین سیستم می‌تواند به همه‌ی فایل‌ها دسترسی داشته باشد. در فایل کانفیگ سرور لیست فایل‌هایی که فقط ادمین‌های سیستم به آن‌ها دسترسی دارند آمده است. در صورتی که از سوی کاربر سیستم دسترسی غیر مجازی انجام شد، پیغام مبنی بر عدم در دسترس بودن فایل به شکل زیر نمایش داده شود.

550: File unavailable.

دستورات

بعد از اینکه کاربر وارد شد می‌تواند 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 می‌دهد:

pwd

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

257: <working directory path>

ساختن فایل یا دایرکتوری جدید

این دستور دایرکتوری جدیدی در مسیر داده شده در سرور ایجاد می کند.

```
mkd <directory path >
```

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

```
257: <directory path> created.
```

پاک کردن فایل یا دایرکتوری

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

```
delete -f <filename>
```

```
delete -d <directory path>
```

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.

```
ls
```

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.
```

عوض کردن نام فایل

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

```
rename <from> <to>
```

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.

```
retr <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 آن ها به کاربر نمایش می دهد.

help

پاسخ این دستور در کانال دستور ارسال می شود 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 می نماید.

quit

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

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 پارامتر های 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

مدیریت حجم کاربران

در فایل کانفیگ سرور حجم مجاز مصرفی^۳ هر کاربر آورده شده است. در هنگام ورود دستور دانلود یک فایل در صورتی که کاربر حجم کافی برای دانلود آن را داشت، فایل را دانلود کرده و مقدار حجم فایل از حجم مجاز کاربر کم می‌شود، و در غیر این صورت پیغامی مبنی بر کافی نبودن حجم کاربر از سرور پاسخ داده می‌شود:

425: Can't open data connection.

کلاینت

شما باید یک کلاینت پیاده سازی کنید تا با استفاده از دستورات گفته شده با سرورتان بتواند ارتباط برقرار کند و این دستورات را انجام دهد.

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

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، فایل‌های که ساخته‌اند یا پاک کرده‌اند و فایل‌هایی که دانلود کرده‌اند.

دقت کنید با هر بار اجرای سرور اگر فایل لاگ وجود نداشت آن را بسازید و در صورتی که از قبل این فایل وجود داشته باشد، در ادامه‌ی آن شروع به نوشتن کنید.

^۳ بر اساس KBytes

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` یا `select() system call` به دلخواه خودتان استفاده کنید)
- برای پیاده‌سازی این تمرین از زبان `C` یا `C++` استفاده کنید.
- ساختار صحیح و تمیزی کد برنامه، بخشی از نمره‌ی این پروژه شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.
- برای هر قسمت کد، گزارش دقیق و شفاف بنویسید. کدهای ضمیمه شده بدون گزارش مربوطه نمره‌ای نخواهند داشت.
- هدف این تمرین یادگیری شماسه است. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، مطابق سیاست درس با گروه متقلب و تقلب دهنده برخورد خواهد شد.
- سوالات 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 دارید، از طریق دو ایمیل زیر 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.

alijavidan@ut.ac.ir ○

mohsenamjadi@ut.ac.ir ○