به نام پروردگار هدایت کننده به راه راست

دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

ترم تحصیلی ۰۳-۰۲

مستند پروژه درس شبکه های کامپیوتری - فاز دوم

استاد درس : دکتر احمدرضا منتظرالقائم

طراحان : امیرعلی گلی، محمدحسین دهقانی، مهرشاد جعفری، مهدی قنبرزاده، محمدحسین رنگ‌رز

توضیحات اولیه

فاز دوم

اف تی پی (FTP) مخفف کلمه File Transfer Protocol به معنی پروتکل انتقال فایل است و عمدتاً برای انتقال فایل بین سرورهای مختلف از طریق اینترنت استفاده می شود. به عنوان مثال، اگر تعدادی فایل دارید که می خواهید آنها را در یک وب سایت آپلود کنید، می توانید از FTP برای انتقال مستقیم فایل ها به سرور اختصاصی میزبان وب سایت استفاده کنید

- پروژه‌ای از این نوع به مهارت‌های شبکه، مدیریت دسترسی، امنیت، و برنامه‌نویسی می‌پردازد. برای افزایش امنیت و قابلیت‌های پیشرفته‌تر، می‌توانید به پیاده‌سازی روش‌های امنیتی توصیه شده برای اتصال FTP مانند FTPS و SFTP نیز بپردازید.
- در این پروژه، از دانشجویان میخواهیم این پروتکل را با استفاده از تجربیات و مطالعات خود در زمینه شبکه در سمت کاربر و سرور با توجه به RFC 959 پیاده سازی کنند

نکات قابل توجه

فاز اول و دوم ، قابل پیاده سازی با زبان های برنامه نویسی java, C++, Python, C# است و برای انجام پروژه شما قادر به تشکیل گروه های **دو نفره** هستید.

مهلت تحویل این فاز تا ۷ دی ماه ساعت ۱۲ شب می‌باشد.

تذکر: توجه داشته باشید شما قادر به استفاده از هیچ کتابخانه ای که قسمتی از پروژه ها را پیاده سازی کرده است نیستید.

فاز دوم: پروتکل FTP

اف تی پی (FTP) یا File Transfer Protocol، یک پروتکل استاندارد برای انتقال فایل‌ها از یک سیستم به سیستم دیگر در شبکه‌های کامپیوتری است. اف تی پی امکان انتقال فایل‌ها به صورت دوطرفه بین یک کلاینت (معمولاً یک کامپیوتر شخصی) و یک سرور (یک سیستم کامپیوتری که فایل‌ها را در اختیار دیگر کامپیوترها قرار می‌دهد) را فراهم می‌کند.

هدف پروژه

هدف اصلی این پروژه در درس شبکه، ایجاد یک سیستم انتقال فایل است که به کاربران امکان می‌دهد فایل‌ها را بین کلاینت و سرور با استفاده از پروتکل اف تی پی انتقال دهند. دانشجویان در این پروژه با مفاهیم مهمی مانند ارتباط شبکه، برنامه‌نویسی سوکت، پروتکل انتقال فایل، امنیت شبکه، احراز هویت و مدیریت دسترسی آشنا می‌شوند و همچنین تجربه کار با ابزارهای شبکه و امنیت را به دست آورند.

قابلیت‌های مدنظر جهت پیاده سازی

در این پروژه، شما باید یک سیستم انتقال فایل طراحی کنید که به کاربران اجازه می‌دهد فایل‌ها را از کلاینت به سرور انتقال دهند. برای آشنایی کامل با این پروتکل،

RFC 959 را مطالعه کنید. برخی از فرمان‌های کاربردی این پروتکل در ادامه توضیح داده می‌شوند.

دستورات

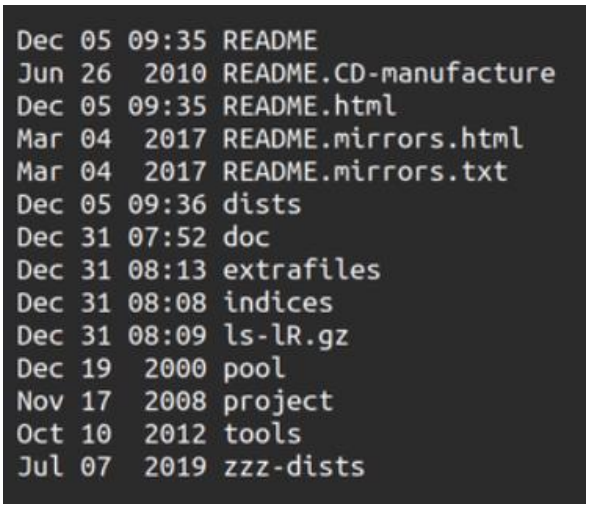
- USER: این فرمان، نام کاربری را برای اتصال به سرور ارسال می‌کند. برای مثال:

USER mahdi

- PASS: پس از وارد کردن نام کاربری، باید رمز عبور نیز ارسال شود. برای مثال:

PASS 1234

- LIST: با استفاده از این فرمان، فایل‌ها و دایرکتوری‌ها به کاربر نشان داده می‌شود. این اطلاعات به صورت یک فهرست به همراه اطلاعاتی مانند نام، اندازه، سطح دسترسی و تاریخ ایجاد است. در شکل 1، نمونه‌ی آن نمایش داده شده است.



شکل 1: اطلاعات فرمان LIST

همچنین کاربر می‌تواند یک پارامتر pathname مشخص کند. در این صورت، اطلاعات مسیر خواسته شده به آن نشان می‌دهد. اگر این مسیر یک دایرکتوری یا گروهی از فایل‌ها باشد، سرور باید فهرست این اطلاعات را به کاربر نمایش دهد. در صورتی که مسیر مشخص شده یک فایل باشد، باید اطلاعات داخل فایل به کاربر نمایش داده شود. به عنوان مثال:

LIST /path/directory

- RETR: برای انتقال یک فایل از سرور به کلاینت، از این فرمان استفاده می‌شود. برای فراخوانی این فرمان نیاز است که مسیر فایل را نیز ارسال کنیم:

RETR /path/file.txt

- STOR: با استفاده از این فرمان، یک فایل را از کلاینت به سرور منتقل می‌کنیم. در صورتی که قبلا این فایل در سمت سرور وجود داشته باشد (نام یکسانی داشته باشند)، فایل جدید جایگزین فایل قبلی می‌شود.

STOR /client-path /server-path

- DELE: یک فایل را می‌توان با مشخص کردن مسیر آن در سمت سرور حذف نمود. زمانی که این فرمان اجرا می‌شود، از سمت سرور یک پیام (Do you really wish to delete? Y/N) برای تایید فرستاده می‌شود و در صورتی که کاربر Y را فشار دهد، فایل حذف خواهد شد.
- MKD: می‌توان یک دایرکتوری جدید در سمت سرور ایجاد کرد. مسیر می‌تواند به دو صورت absolute و relative مشخص شود.
در صورتی که مسیر absolute باشد (نسبت به روت مشخص می‌شود):
MKD /home/user
زمانی که مسیر relative است (نسبت به مسیر فعلی مشخص می‌شود):
MKD ../folder
- RMD: می‌توان یک دایرکتوری را در سمت سرور حذف نمود. مطابق فرمان KMD، می‌توان مسیر را به دو صورت absolute و relative مشخص کرد.
- PWD: این فرمان مسیر فعلی در سمت سرور را نمایش می‌دهد. زمانی که PWD اجرا می‌شود، مسیر فعلی به کاربر (کلاینت) نمایش داده می‌شود:

/home/user/public

- CWD: می‌توان مسیر را در سمت سرور عوض کرد. این کار نیز به دو صورت absolute و relative انجام می‌شود. برای مثال:

CWD /home

- CDUP: مشابه دستور CWD، برای عوض کردن مسیر فعلی در سرور استفاده می‌شود. با این تفاوت که با اجرا شدن این دستور، به دایرکتوری والد منتقل می‌شویم. برای مثال، اگر دستور CDUP در مسیر /home/user/public اجرا شود، مسیر جدید /home/user خواهد بود.

- QUIT: این دستور باعث می‌شود که ارتباط کلاینت یا سرور قطع شود. در صورتی که یک جابه‌جایی فایل در حال انجام باشد، ابتدا عملیات کامل شده و سپس ارتباط بسته می‌شود.
- در صورتی که این عملیات با موفقیت انجام شود و دایرکتوری وجود داشته باشد، پیغام
200 directory changed to /home نمایش داده می‌شود. در غیر این صورت خطای
400 directory doesn’t exist نشان داده می‌شود. برای بقیه فرمان‌ها می‌توان از این پاسخ ایده گرفت. برای مثال، زمانی که نام کاربری و رمز عبور کاربر درست است، پیغام 200 (عملیات موفق)، در غیر این صورت 400 (خطا) به همراه یک متن مناسب نمایش داده شود.

این پروژه شامل قابلیت‌های زیر است:

- **سرور:** شما باید یک سرور اف تی پی ایجاد کنید که فایل‌ها و دایرکتوری‌ها را نگهداری می‌کند و همچنین باید کلاینت‌ها به آن متصل شوند و فرمان‌های مشخص شده را اجرا کنند. زمانی که اتصال کلاینت برقرار می‌شود، یک لیست از دستورات مثل LIST، RETR و... به همراه نحوه استفاده به کلاینت فرستاده می‌شود.
- **کلاینت:** چند برنامه کلاینت که به کاربران امکان می‌دهند وارد سرور شوند، فایل‌ها را مشاهده کنند و همچنین فایل‌ها را دریافت کنند.
- **فرمان‌های موجود:** کلاینت باید بتواند فایل‌های خود را با فرمان STOR به سرور بفرستد، با فرمان RETR یک فایل را دریافت کند و همچنین از دیگر فرمان‌های توضیح داده شده مانند LIST، DELE، MKD، RMD، PWD، CDUP و QUIT پشتیبانی کند.
- **مدیریت دسترسی‌ها:** دسترسی‌های کاربران باید مشخص شود. برای مثال، برخی از فایل‌ها و دایرکتوری‌ها به صورت خصوصی (private) در سمت سرور تعریف شده‌اند و همه‌ی کاربران دسترسی به این فایل‌ها و دایرکتوری‌ها را ندارند و فقط کاربران مشخص شده قابلیت خواندن و نوشتن را خواهند داشت.
- **مدیریت خطا:** برنامه‌ی شما باید خطا را به درستی مدیریت کند و پیغام مناسب به کاربر نشان دهد. برای مثال، زمانی که فرمان CWD /home اجرا می‌شود، در صورتی که این عملیات با موفقیت انجام شود و دایرکتوری وجود داشته باشد،

پیغام 200 directory changed to /home نمایش داده می‌شود. در غیر این صورت خطای 400 directory doesn’t exist نشان داده می‌شود. برای دیگر فرمان‌ها نیز می‌توان از این پاسخ ایده گرفت. برای مثال، زمانی که نام کاربری و رمز عبور درست است، پیغام 200 (عملیات موفق)، در غیر این صورت 400 (خطا) به همراه یک متن مناسب نمایش داده شود.

- **گزارش‌گیری:** تمامی عملیات‌ها و فعالیت‌ها در سمت کلاینت و سرور برای افزایش امنیت باید در سمت سرور ثبت شوند و کلاینت ها با دستور REPORT بتوانند آن را مشاهده کنند. در واقع این فرمان تاریخچه فرمان‌های اجرا شده در سمت سرور را به همه نمایش می‌دهد. (می‌توان صرفا یک کاربر مشخص به عنوان ادمین به این اطلاعات دسترسی داشته باشد).
- **امنیت:** در مورد روش‌های تقویت امنیت اتصال اف تی پی مانند اف تی پی اس (FTPS)و اس اف تی پی (SFTP) مطالعه کنید (اجباری) و در صورت امکان برخی از آن‌ها را در برنامه خود پیاده‌سازی کنید (امتیازی).