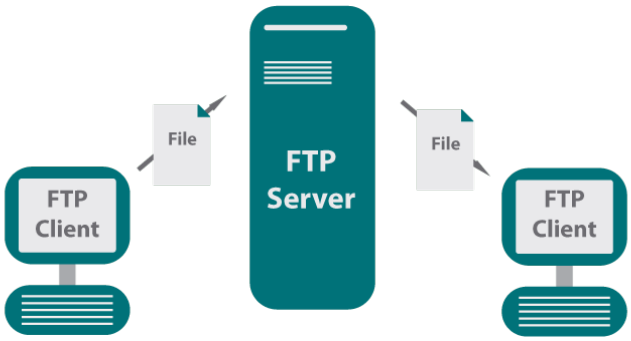
۲.۰ فاز دوم :پیاده‌سازی پروتکل FTP



پروتکل انتقال فایل [۱۸] یا به اختصار FTP ، یک پروتکل شبکه است که برای انتقال فایل‌ها بین سرورها و سیستم‌های مختلف از طریق اینترنت طراحی شده است. این پروتکل در بسیاری از کاربردهای اینترنتی مورد استفاده قرار می‌گیرد و به کاربران این امکان را می‌دهد که به راحتی فایل‌ها را از یک دستگاه به دستگاه دیگر منتقل کنند. به عنوان مثال، زمانی که شما تعدادی فایل دارید که قصد دارید آنها را به وب‌سایت خود اضافه کنید؛ FTP ابزاری بسیار مفید برای این کار محسوب می‌شود. با استفاده از FTP، کاربران می‌توانند به سرور میزبان وب‌سایت خود متصل شوند و فایل‌های مورد نظر خود را به صورت مستقیم و سریع به سرور آپلود کنند.

FTP امکان انتقال فایل‌ها با سرعت بالا و به صورت دسته‌ای را فراهم می‌سازد و می‌تواند برای دانلود فایل‌ها از سرور نیز مورد استفاده قرار گیرد. FTP می‌تواند با استفاده از امنیت افزوده با پروتکل‌های مانند FTPS [۱۹] یا SFTP [۲۰] ترکیب شود تا امنیت انتقال داده‌ها را افزایش دهد.

برای اطلاعات بیشتر به RFC ۷۶۵ مراجعه کنید

[۱۸] File Transfer Protocol [۱۹] File Transfer Protocol Secure [۲۰] secure File Transfer Protocol

۱.۲.۰ هدف پروژه

هدف اصلی این پروژه در درس شبکه، توسعه یک سیستم انتقال فایل است که به کاربران امکان می‌دهد تا فایل‌ها را بین کلاینت و سرور با استفاده از پروتکل FTP منتقل کنند. دانشجویان در این پروژه با مفاهیم کلیدی مانند ارتباطات شبکه، برنامه‌نویسی سوکت، پروتکل انتقال فایل، امنیت شبکه، احراز هویت و مدیریت دسترسی آشنا می‌شوند. همچنین تجربه عملی در کار با ابزارهای شبکه و امنیت را به دست خواهند آورد. این پروژه نه تنها به تقویت مهارت‌های فنی دانشجویان کمک می‌کند، بلکه باعث درک عمیق و بهتری از ساختار و عملکرد سیستم‌های انتقال فایل و چالش‌های امنیتی مرتبط با آن می‌شود.

توضیحات دستورات FTP

۱. **USER**

- این دستور برای ارسال نام کاربری به سرور به کار می‌رود. سرور از این طریق می‌داند که کدام کاربر می‌خواهد وارد سیستم شود.
- **کدهای وضعیت:**
  - ۳۳۱: نام کاربری پذیرفته شده است، انتظار وارد کردن رمز عبور.
  - ۵۳۰: ورودی نامعتبر؛ لطفاً دوباره تلاش کنید.

۲. **PASS**

- این دستور برای ارسال رمز عبور به سرور استفاده می‌شود. پس از آن، سرور اعتبار کاربر را بررسی می‌کند.
- **کدهای وضعیت:**
  - ۲۳۰: ورود موفقیت‌آمیز.
  - ۵۳۰: رمز عبور نامعتبر؛ لطفاً دوباره تلاش کنید.

LIST .۳

- این دستور لیستی از فایل‌ها و دایرکتوری‌های موجود در دایرکتوری فعلی سرور را برای کلاینت ارسال می‌کند. این اطلاعات به صورت یک فهرست به همراه اطلاعاتی مانند نام، اندازه، سطح دسترسی و تاریخ ایجاد است.
- **کدهای وضعیت:**
  - ۱۲۵: در حال آماده‌سازی انتقال لیست.
  - ۲۲۶: لیست با موفقیت منتقل شد.

```
Dec 05 09:35 README
Jun 26  2010 README.CD-manufacture
Dec 05 09:35 README.html
Mar 04  2017 README.mirrors.html
Mar 04  2017 README.mirrors.txt
Dec 05 09:36 dists
Dec 31 07:52 doc
Dec 31 08:13 extrafiles
Dec 31 08:08 indices
Dec 31 08:09 ls-lR.gz
Dec 19  2000 pool
Nov 17  2008 project
Oct 10  2012 tools
Jul 07  2019 zzz-dists
```

همچنین کاربر می‌تواند یک پارامتر `pathname` مشخص کند. در این صورت، اطلاعات مسیر خواسته‌شده به کاربر نشان‌داده می‌شود. اگر این مسیر یک دایرکتوری یا گروهی از فایل‌ها باشد، سرور باید فهرست این اطلاعات را به کاربر نشان‌دهد.

```
LIST /path/to/directory
```

RETR .۴

- این دستور برای دانلود یک فایل از سرور استفاده می‌شود.

```
RETR /path/to/file
```

- **کدهای وضعیت:**
  - ۱۵۰: در حال آماده‌سازی انتقال فایل.
  - ۲۲۶: انتقال فایل با موفقیت انجام شد.

۵. **STOR**

- این دستور برای آپلود یک فایل به سرور استفاده می‌شود. در صورتی که قبلا این فایل در سمت سرور وجود داشته‌باشد (نام یکسانی داشته باشند)، فایل جدید جایگزین فایل قبلی می‌شود.

```
STOR /client-path /server-path
```

- **کدهای وضعیت:**
  - ۱۵۰: در حال آماده‌سازی برای دریافت فایل.
  - ۲۲۶: فایل با موفقیت بارگذاری شد.

۶. **DELE**

- این دستور برای حذف یک فایل از سرور به کار می‌رود.

```
DELE /path/to/file
```

- **کدهای وضعیت:**
  - ۲۵۰: حذف فایل با موفقیت انجام شد.
  - ۵۵۰: فایل موجود نیست یا نمی‌توان آن را حذف کرد.

۷. **MKD**

- این دستور برای ایجاد یک دایرکتوری (پوشه) جدید در سرور استفاده می‌شود.

```
MKD /path/to/new/dir
```

- **کدهای وضعیت:**
  - ۲۵۷: دایرکتوری جدید با موفقیت ایجاد شد.
  - ۵۵۰: نمی‌توان دایرکتوری ایجاد کرد.

RMD .۸

- این دستور برای حذف یک دایرکتوری از سرور استفاده می‌شود.

```
RMD /path/to/dir
```

- **کدهای وضعیت:**
  - ۲۵۰: دایرکتوری با موفقیت حذف شد.
  - ۵۵۰: دایرکتوری وجود ندارد یا نمی‌توان آن را حذف کرد.

PWD .۹

- این دستور برای دریافت مسیر دایرکتوری فعلی سرور به کار می‌رود.
- **کدهای وضعیت:**
  - ۲۵۷: مسیر دایرکتوری فعلی با موفقیت نمایش داده شد.

CWD .۱۰

- این دستور برای تغییر دایرکتوری فعلی به دایرکتوری مشخص‌شده استفاده می‌شود.

```
CWD /dir
```

- **کدهای وضعیت:**
  - ۲۵۰: تغییر دایرکتوری با موفقیت انجام شد.
  - ۵۵۰: دایرکتوری موجود نیست.

CDUP .۱۱

- این دستور برای تغییر به دایرکتوری والد استفاده می‌شود. برای مثال، اگر دستور

```
CDUP
```

در مسیر `/home/user/public` اجرا شود، مسیر جدید `/home/user` خواهد بود.

- **کدهای وضعیت:**
  - ۲۵۰: تغییر به دایرکتوری والد با موفقیت انجام شد.

QUIT .۱۲

- این دستور برای قطع ارتباط کاربر با سرور استفاده می‌شود. اگر فایلی در حال انتقال باشد نباید تداخلی برای آن پیش بیاید.
- **کدهای وضعیت:**
  - ۲۲۱: ارتباط با موفقیت قطع شد.

۲.۲.۰ توضیحات تکمیلی

مسیرهای نسبی و مطلق

در دستورات FTP که با مسیرها سروکار دارند (مانند دستورات CWD، MKD، RMD، RETR، STOR و ...)، مسیرها می‌توانند به دو صورت نسبی و مطلق تعریف شوند:

- **مسیر نسبی** [۲۱]: مسیری که نسبت به دایرکتوری فعلی تعیین می‌شود. برای مثال، اگر کاربر در دایرکتوری /home/user/ باشد و بخواهد به زیرشاخه documents/ دسترسی پیدا کند، می‌تواند مسیر نسبی documents/ را استفاده کند.
- **مسیر مطلق** [۲۲] : مسیری که از ریشه سیستم فایل سرور آغاز می‌شود و به‌طور کامل مشخص‌کننده مسیر است. برای مثال، /home/user/documents/ یک مسیر مطلق است.

مدیریت خطا

برای هر دستور FTP، مدیریت خطا [۲۳] به منظور اطمینان از اجرای صحیح عملیات ضروری است. در صورتی که مشکلی در اجرای یک دستور به وجود بیاید، سرور با استفاده از کدهای وضعیت مشخص، نوع خطا را به کاربر اطلاع می‌دهد. به عنوان مثال:

- اگر کاربر بخواهد به یک دایرکتوری غیرموجود برود، دستور CWD کد وضعیت ۵۵۰ را برمی‌گرداند که نشان‌دهنده این است که دایرکتوری وجود ندارد.
- هنگام حذف یا انتقال فایل، اگر فایل مورد نظر پیدا نشود یا دسترسی کافی برای این عملیات وجود نداشته باشد، سرور پیام مناسب خطا را ارسال می‌کند.

---

[۲۱] Relative [۲۲] Absolute [۲۳] Error Handling

مدیریت انتقال فایل‌ها با استفاده از کانکشن مجزا

در پروتکل FTP، انتقال داده‌ها (مانند آپلود یا دانلود فایل‌ها) از طریق یک کانکشن مجزا به نام اتصال داده انجام می‌شود. در واقع، هنگام اجرای دستورات انتقال فایل مانند RETR و STOR:

- سرور و کلاینت ابتدا یک **اتصال کنترل** [۲۴] برای ارسال دستورات و پاسخ‌ها ایجاد می‌کنند.
- برای انتقال واقعی فایل‌ها، یک اتصال جداگانه به نام **اتصال داده** [۲۵] باز می‌شود. این اتصال به طور موقت ایجاد شده و پس از اتمام عملیات انتقال، بسته می‌شود.

این روش به سرور و کلاینت اجازه می‌دهد که دستورات و فایل‌ها را به طور مستقل و همزمان مدیریت کنند و از تداخل بین اتصال کنترل و اتصال داده جلوگیری می‌کند. این ویژگی در FTP یکی از دلایل اصلی انعطاف‌پذیری و کارایی آن است، به خصوص در انتقال فایل‌های بزرگ یا استفاده چندگانه از دستورات مختلف به‌صورت همزمان.

سرور و کلاینت در سیستم انتقال فایل

در سیستم انتقال فایل مبتنی بر پروتکل FTP، دو جزء اصلی وجود دارد که هر کدام وظایف خاصی را بر عهده دارند:

۱. سرور [۲۶]

سرور فایل، نقش اصلی را در مدیریت فایل‌ها و دایرکتوری‌ها بازی می‌کند. این سرور به درخواست‌های کلاینت‌ها پاسخ می‌دهد و وظایف زیر را بر عهده دارد:

- ذخیره و مدیریت فایل‌ها و دایرکتوری‌ها در سیستم.
- احراز هویت کاربران از طریق اعتبارسنجی نام کاربری و رمز عبور.
- کنترل دسترسی به فایل‌ها و دایرکتوری‌ها با توجه به مجوزها و سطح دسترسی تعریف‌شده.
- ایجاد، حذف و تغییر دایرکتوری‌ها و فایل‌ها طبق درخواست‌های مجاز.
- ارسال یا دریافت فایل‌ها از/به کلاینت‌ها با استفاده از اتصال‌های داده

[۲۴] Control Connection [۲۵] Data Connection [۲۶] Server

۲. کلاینت [۲۷]

کلاینت سیستم به کاربران این امکان را می‌دهد تا با سرور فایل ارتباط برقرار کنند و عملیات‌های مختلفی را انجام دهند:

- ارسال درخواست‌ها به سرور برای دسترسی به فایل‌ها یا دایرکتوری‌ها.
- مدیریت فایل‌ها از طریق دستورات FTP مانند STOR، RETR، DELE یا CWD.
- تعامل با سرور برای مدیریت انتقال فایل‌ها، احراز هویت، و انجام عملیات‌های مختلف.

مدیریت دسترسی‌ها

یکی از بخش‌های حیاتی در سیستم FTP، مدیریت دسترسی‌ها [۲۸] است که از اهمیت بالایی برخوردار است. سرور باید بتواند سطح دسترسی‌های مختلف را به درستی مدیریت کند تا اطمینان حاصل شود که کاربران تنها به فایل‌ها و دایرکتوری‌هایی که مجوز آن‌ها را دارند دسترسی پیدا کنند. مدیریت دسترسی‌ها شامل موارد زیر می‌شود:

۱. احراز هویت [۲۹]

زمانی که کاربر با استفاده از دستورات USER و PASS به سرور متصل می‌شود، سرور باید اطلاعات ورود کاربر را اعتبارسنجی کند. این فرآیند احراز هویت از طریق بررسی نام کاربری و رمز عبور انجام می‌شود. کاربری که نتواند احراز هویت کند، دسترسی به سرویس‌ها و منابع FTP نخواهد داشت.

---

۲۷ Client ۲۸ Access Control ۲۹ Authentication

۲. سطوح دسترسی [۳۰]

پس از احراز هویت، دسترسی کاربر به منابع سرور محدود به سطوح دسترسی تعریف‌شده خواهد بود. این سطوح دسترسی می‌توانند شامل موارد زیر باشند:

- **دسترسی خواندن** [۳۱]: کاربر می‌تواند فایل‌ها را مشاهده و دانلود کند اما نمی‌تواند تغییراتی در آن‌ها ایجاد کند.
- **دسترسی نوشتن** [۳۲]: کاربر می‌تواند فایل‌ها را آپلود کرده یا ویرایش کند.
- **دسترسی حذف** [۳۳]: کاربر مجاز است فایل‌ها و دایرکتوری‌ها را حذف کند.
- **دسترسی ایجاد** [۳۴]: کاربر می‌تواند دایرکتوری‌های جدید ایجاد کند.

---

۳۰ Permissions ۳۱ Read ۳۲ Write ۳۳ Delete ۳۴ Create

امنیت

رمزنگاری اطلاعات در پروتکل‌های مختلف شبکه‌ای اهمیت بسیاری دارد. دو پروتکل اصلی که برای این منظور استفاده می‌شوند، SSL و SSH هستند. هر یک از این پروتکل‌ها با رویکردهای متفاوتی عمل می‌کنند، اما هدف اصلی هر دو، ایمن‌سازی ارتباطات شبکه‌ای و جلوگیری از شنود یا دسترسی غیرمجاز به اطلاعات در حال انتقال است.عنوان مثال در پروتکل FTP به هنگام ارسال نام کاربری و رمز عبور یک حمله مرد میانی می‌تواند این داده‌ها را شنود کرده و به سرقت ببرد. به همین منظور دو پروتکل دیگر برای ارتقای امنیت FTP عرضه شده است.

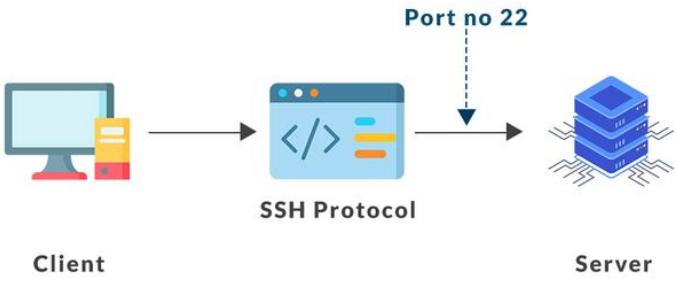
FTPS

FTPS یا Secure FTP نسخه‌ای امن از پروتکل FTP است که از TLS [۳۵] یا SSL [۳۶] برای رمزنگاری انتقال داده‌ها استفاده می‌کند. این پروتکل به منظور افزایش امنیت FTP طراحی شده است تا اطلاعات حساس مانند نام کاربری و رمز عبور و همچنین فایل‌ها، در طول انتقال به صورت رمزنگاری‌شده ارسال شوند.

[۳۵] Transport Layer Security

[۳۶] Secure Socket Layer

SFTP



SFTP مخفف SSH File Transfer Protocol است. این پروتکل برای انتقال فایل به شکل امن طراحی شده و از SSH [۳۷] استفاده می‌کند تا داده‌ها را در طول مسیر رمزنگاری کند. SFTP یک پروتکل کاملاً متفاوت از FTP و FTPS است و به خاطر امنیت و سهولت استفاده‌اش در بسیاری از موارد جایگزین FTP شده است. به طور خلاصه، تفاوت اصلی این دو پروتکل در این است که SSL بیشتر برای ایمن‌سازی ارتباطات وب مانند HTTPS استفاده می‌شود، در حالی که SSH بیشتر برای ایمن‌سازی ارتباطات در سطوح پایین‌تری مانند دسترسی امن به سرورها به کار می‌رود. هر دوی این پروتکل‌ها از روش‌های پیچیده‌ای برای رمزنگاری اطلاعات استفاده می‌کنند و ساختار و معماری آن‌ها متفاوت است.

پروتکل SSL لایه‌ای بر روی پروتکل‌های استاندارد شبکه مانند HTTP ایجاد می‌کند تا اطلاعات در حین انتقال میان کلاینت و سرور به صورت رمزنگاری‌شده ارسال شود. در مقابل، SSH یک پروتکل جامع‌تر برای دسترسی امن به سیستم‌های راه دور است که از رمزنگاری به عنوان بخشی از ارتباطات کلی خود استفاده می‌کند. شما باید با انتخاب یکی از این دو پروتکل عملیات رمزنگاری را بر روی اطلاعات ارسالی انجام دهید.

---

[۳۷] Secure Shell