

شبکه های کامپیوتری

کیاوش جمشیدی ۸۱۰۱۹۷۴۸۶

سینا سلیمیان ۸۱۰۱۹۷۵۲۸

برای برقراری ارتباط بین کلاینت و سرور از سوکت استفاده میکنیم و چون قرار است روی ۲ پورت اطلاعات را منتقل کنیم برای هر دو، دو سوکت در نظر میگیریم. یکی را روی پورت `command channel port`، `bind` میکنیم و دیگری را روی `data channel port`.

سیستم کال `select` : برای برقراری ارتباط همزمان سرور و چند تا کلاینت از این `system call` استفاده میکنیم به طوری که با هر درخواست جدید از سمت هر کدام از کلاینت ها در همان لحظه سرور متوجه می شود و به سراغ هندل کردن آن دستور می رود.

(در تابع `handle commands` )

**Handle commands** : در این تابع ابتدا دستور داده شده از سمت کلاینت را `tokenize` میکنیم و با توجه به توکن اول آن تابع مربوطه را `call` می کنیم به شکل زیر :

#### Help

همه دستورات را در این تابع با توضیحات چاپ میکنیم.

#### Pwd

در این قسمت ما دایرکتوری فعلی کلاینت را چاپ میکنیم.

#### Cwd

اگر بدون آرگومان باشد باید به دایرکتوری اولیه کلاینت برویم در غیر این صورت به مسیر داده شده دایرکتوری را عوض میکنیم.

#### Mkd

در این قسمت در مسیر فعلی کلاینت ما یک دایرکتوری جدید تولید میکنیم.

#### Dele -f

اگر این فایل داده شده وجود نداشته بود، `error` میدهیم در غیر این صورت فایل را پاک میکنیم ( اگر کلاینت ادمین نبود ولی فایل جزو فایل هایی که فقط ادمین بهش دسترسی دارد باشد باز هم `file unavailable` میدهیم )

#### Dele -d

اگر دایرکتوری موجود باشد آنرا پاک میکنیم در غیر این صورت `error` می دهیم.

## Ls

در این قسمت همه فایل ها و دایرکتوری های داهل مسیر کنونی کلاینت را چاپ میکنیم و از طریق پورت دیتا آنرا میفرستیم.

## Rename

اگر فایل موجود نبود error می‌دهیم. اگر برای ادمین باشد و کلاینت ادمین نبود file unavailable می‌دهیم. در غیر این صورت فایل را rename میکنیم.

## Retr

اگر حجم کافی نداشته باشیم cant open data connection می‌دهیم. اگر برای ادمین باشد و کلاینت ادمین نباشد file unavailable می‌دهیم. اگر هم شرایط برقرار باشد آنرا دانلود و از طریق پورت دیتا آنرا میفرستیم و چاپ میکنیم.

## Quit

در این تابع ما کاربر فعلی را logout می‌کنیم.

در همه این حالات اگر لاگین نکرده باشیم error می‌دهیم که باید حتما وارد شده باشیم.

اگر کاربری وارد نشده باشد و رمز را وارد کنیم bad sequence of commands می‌دهیم.

با استفاده از کتابخانه json.hpp ما config.json را خوانده و کلاس user های مربوطه را درست می‌کنیم.

دو تا آرایه sockfds و sockfd\_usernames داریم که در ایندکس های یکسان username و sockfd کلاینت را به ما می‌دهد.

تابع find username sockfd اینگونه عمل میکند که به آن sockfd را می‌دهیم و نام کاربری user را می‌دهد.

تابع find user by sockfd اینگونه عمل میکند که به آن sockfd را می‌دهیم و یک instance از کلاس user را می‌دهد.

در صورت موفقیت آمیز بودن دستور وارد شده ما لحظه آن command را ثبت کرده و با توضیحات در log\_file.txt آنرا شرح و append می‌دهیم.

تابع write to client هم به این صورت است که با سیستم کال write در sockfd کلاینت مورد نظر دستور فعلی را ارسال می‌کند و در سمت کلاینت آنرا read می‌کند.