

Projeto de Redes de Computadores

Trabalho de implementação de aplicação em redes utilizando sockets. IC/UFAL 2020.1.

Alunos

- João Victor Falcão Santos Lima
- Rodrigo Santos da Silva

Principais Funcionalidades

A aplicação segue um modelo Cliente/Servidor, onde o seu objetivo é realizar a transferência de arquivos entre os sistemas finais. Nela, um hospedeiro deve ser o servidor centralizado, que armazenará os arquivos e atenderá requisições de múltiplos clientes, os quais poderão enviar novos arquivos ao servidor, ou baixar os que já estão lá. Assim, a aplicação possui 3 funções principais, sendo elas “1 - Listar arquivos no servidor”, “2 - Enviar arquivo”, e “3 - Baixar arquivo”.

A primeira funcionalidade, como o próprio nome diz, tem a função de apresentar todos os arquivos que estão localizados no servidor. A segunda funcionalidade tem a função de transmitir um arquivo localizado no cliente que se conectou com o servidor. Por último, a terceira funcionalidade tem a função de realizar download de um dos arquivos localizados no servidor. Além disso, a aplicação possui uma funcionalidade básica, semelhante a um logoff, para encerrar formalmente a comunicação com o servidor antes de finalizar a execução do cliente.

O que poderia ser implementado a mais

Inicialmente, outras funções básicas como Deletar um arquivo, Sobrecrever um arquivo existente no servidor e realizar o download ou envio de vários arquivos em uma única requisição. Além disso, uma interface gráfica poderia ter sido desenvolvida, para que o usuário utilizasse a aplicação de forma mais intuitiva, e a disponibilização (deploy) da aplicação também poderia ter ocorrido.

Dificuldades Encontradas

Algumas funções primitivas na utilização dos Sockets, como `recv()` e `send()` apresentaram alguns erros no processo de desenvolvimento, como não encontrar o arquivo que você apresenta e o estabelecimento das conexões, mas nada que não fosse resolvido lendo a documentação e verificando como essas funções funcionavam.

Instruções para Execução

A aplicação foi desenvolvida na linguagem de programação Python. Assim, é necessário ter uma versão recente do Python instalada (e.g. 3.9.x). Além disso, não foram utilizadas bibliotecas ou

módulos externos, logo não é necessário fazer a instalação de nenhuma dependência.

Para executar a aplicação, primeiramente deve-se executar o servidor. Para isso, utilize o terminal para executar o script `server.py` que está na pasta `server`. Por padrão, será criado um socket na porta 5050.

```
cd server
python3 server.py
```

Esse script aceita um parâmetro adicional `--port` para definir outro número de porta caso desejado. Nesse caso, o comando seria algo semelhante a:

```
python3 server.py --port 3333
```

Com o servidor em execução, devemos utilizar um outro terminal para executar o cliente, que é o script `client.py` na pasta `client`. Podemos utilizar vários terminais para executar instâncias de clientes diferentes.

```
cd client
python3 client.py
```

Por padrão, o cliente tentará criar um socket para o endereço IP 127.0.0.1 (localhost) na porta 5050. Mas o script `client.py` também aceita dois parâmetros adicionais: `--port` para especificar outro número de porta (caso a porta do servidor tenha sido alterada), e `--ip`, para especificar outro endereço IP caso o servidor esteja em execução em outro computador da rede local. Por exemplo:

```
python3 client.py --ip 192.168.0.40 --port 3333
```

Em cada cliente, será exibido um menu listando as opções para cada funcionalidade. A entrada deve ser o número correspondente à funcionalidade desejada. Em seguida, serão exibidas mensagens que instruirão o usuário a realizar as próximas entradas necessárias. Note que há um tratamento para arquivos com nome repetido: eles não serão sobrescritos, mas sim uma cópia será salva com um prefixo indicando seu número (e.g. arquivo `ola.txt` e as cópias `1-ola.txt` e `2-ola.txt`). No lado do servidor, os arquivos serão salvos no diretório `server/files`, já no cliente eles serão salvos na própria pasta `client`.