

COMPARTILHAMENTO DE ARQUIVOS P2P

Guilherme Magnus Felipe de Oliveira
Rafaela Aparecida Pereira Paiva

CLIENTE TCP...

- Obtém os endereços de IP dos servidores participantes da rede;

```
arquivos = open('arquivo_configuracao.txt', 'r')  
texto = arquivos.read()
```

- O cliente solicita o arquivo que deseja e Recebe como resposta o hash do arquivo, caso não tenha, recebe “None”;

CLIENTE TCP...

```
pesquisaArquivo = raw_input('Digite o Arquivo A Ser Pesquisado ou sair para finalizar o programa:')

if(pesquisaArquivo == "sair"):
    break

listHashRecebidos = []
for ip in listServerName:
    if ip == '':
        continue
    clientSocket = socket(AF_INET, SOCK_STREAM)
    clientSocket.connect((ip, serverPort))
    clientSocket.send(pesquisaArquivo.encode('utf-8'))
    listHashRecebidos.append('download:' + clientSocket.recv(1024).decode('utf-8'))
    clientSocket.close()
```

CLIENTE TCP...

- Escolhe de qual servidor deseja fazer o download do arquivo;

```
indiceHash = int(text)
if indiceHash < 0 or indiceHash >= len(listHashRecebidos):
    continue

hashEscolhido = listHashRecebidos[indiceHash]
ipEscolhido = listServerName[indiceHash]
```

CLIENTE TCP...

- Faz uma nova conexão com o endereço IP do hash escolhido;

```
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((ipEscolhido, serverPort))
clientSocket.send(hashEscolhido.encode('utf-8'))
```

CLIENTE TCP...

- Recebe o arquivo solicitado.

```
arquivo = open(diretorioPadrao+pesquisaArquivo, 'wb')

dados = clientSocket.recv(4096)

# Le os dados
while(dados):
    # Escreve os dados do arquivo
    arquivo.write(dados)

    # Recebe os dados do arquivo
    dados = clientSocket.recv(4096)

# Fecha o arquivo
arquivo.close()

# Finaliza a conexao
clientSocket.close()

print("Transferencia concluida!")
```

SERVIDOR TCP...

- Escolhe o diretório a ser compartilhado e abre uma conexão;

```
diretorioPadrao = "./pasta_compartilhada/"
arquivosEncontrados = {}

def iniciarServidor():
    serverSocket = socket(AF_INET, SOCK_STREAM)
    serverSocket.bind(('', serverPort))
    serverSocket.listen(10)
```

SERVIDOR TPC...

- Recebe o nome do arquivo que um cliente deseja;

```
def ouvirSocket(connectionSocket, addr):  
    buscaDownload = connectionSocket.recv(1024)  
    buscaDownload = buscaDownload.decode('utf-8')
```


SERVIDOR TCP...

- Pesquisa o arquivo no diretório;

```
def pesquisaArquivo( diretorioPadrao, buscaDownload ):
    pathName = diretorioPadrao
    for path, diretorio, arquivos in os.walk(pathName):
        for fileName in arquivos:
            if fileName == buscaDownload:
                arquivo = open("{}{}".format(path, fileName), 'r')
                return arquivo
    print("Arquivo Não Encontrado")
    return None
```

SERVIDOR TCP...

- Gera o hash do arquivo e envia o hash ao cliente;

```
if not buscaDownload.__contains__('download:'):
    arquivo = pesquisaArquivo(diretorioPadrao, buscaDownload)
    hash = criarHash(arquivo)
    hash = str(hash)
    arquivosEncontrados[hash] = arquivo
    connectionSocket.send(hash.encode('utf-8'))
print("Solicitado arquivo por: {}".format(addr))
```

SERVIDOR TCP...

- O servidor recebe do cliente uma nova conexão solicitando o envio do arquivo;

```
else:
    hash = buscaDownload.replace('download:', '')

# acessa o dicionario de hash com o hash que quer fazer o download.
    arquivo = arquivosEncontrados[hash]
    arquivo = open(arquivo.name, 'rb')
dado = arquivo.read(4096)
while(dado):
    connectionSocket.send(dado)
    dado = arquivo.read(4096)

    print("Arquivo enviado com sucesso para: {}".format(addr))
```

ENCERRANDO CONEXÃO...

