Atividade 1 - Programação com Sockets TCP
Sistemas Distribuídos – Prof. Rodrigo Campiolo

Questão 1) Faça um servidor para processar as seguintes mensagens dos clientes. O servidor deve suportar mensagens de múltiplos clientes. Use o TCP. As mensagens de solicitação estão no formato String UTF:

CONNECT user, password
* Conecta o usuário **user** a sua área no servidor se a senha **password** conferir e o usuário **user** existir. Em caso de sucesso, devolver **SUCCESS** como String UTF. Em caso de falha, devolver **ERROR**. Obs: *password* deve ser enviado o *hash* em SHA-512.

PWD
* Devolve o caminho corrente (PATH) usando String UTF separando os diretórios por barra(/).

CHDIR *path*
* Altera o diretório corrente para **path**, retornado uma String UTF, **SUCCESS**, no caso de sucesso, e **ERROR**, no caso de erro.

GETFILES
* Devolve os arquivos do diretório corrente no servidor.
Formato:
    * devolve um inteiro indicando o número de arquivos como String UTF.
    * devolve cada arquivo como uma String UTF.

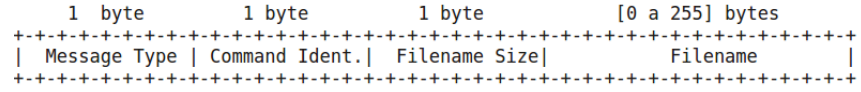
GETDIRS
* Devolve os diretórios do diretório corrente no servidor.
Formato:
    * devolve um inteiro indicando o número de diretórios como String UTF.
    * devolve cada diretório como uma String UTF.

EXIT
* Finaliza a conexão

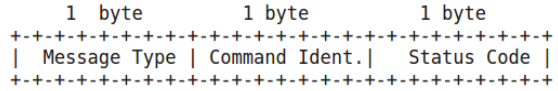
Questão 2) Faça uma aplicação com um servidor que gerencia um conjunto de arquivos remotos entre múltiplos usuários. O servidor deve responder aos seguintes comandos:
-> ADDFILE (1): adiciona um arquivo novo.
-> DELETE (2): remove um arquivo existente.
-> GETFILESLIST (3): retorna uma lista com o nome dos arquivos.
-> GETFILE (4): faz download de um arquivo.

O servidor deve registrar as ações em *logs*. Use uma biblioteca da linguagem específica para manipular os *logs*.

As solicitações possuem o seguinte cabeçalho em comum:
1 byte: requisição (1) – *Message Type (0x01)*
1 byte: código do comando – *Command Identifier (0x01 a 0x04)*
1 byte: tamanho do nome do arquivo – *Filename Size*
variável [0-255]: nome do arquivo em bytes – *Filename*



As respostas possuem o seguinte cabeçalho em comum:
1 byte: resposta (2) – *Message Type (0x02)*
1 byte: código do comando – *Command Identifier (0x01 a 0x04)*
1 byte: status code (1-SUCCESS, 2-ERROR) – *Status Code*



---
para o **ADDFILE**, adicionam-se os campos na solicitação:
4 bytes: tamanho do arquivo (*big endian order*) em bytes.
variável[1 a 2^{32}]: bytes do arquivo.

---
para o **GETFILESLIST**, adicionam-se os campos na resposta:
2 bytes: número de arquivos (*big endian order*)
Repete-se até terminar os nomes:
    1 byte: tamanho do nome (1-255)
    variável [1 a 255]: nome do arquivo
---
para o **GETFILE**. adicionam-se os campos na resposta:
4 bytes: tamanho do arquivo (big endian order) em bytes
variável [1 a 2^{32}]: bytes do arquivo.

* ao fazer download do arquivo, grave em uma pasta padrão.
* para enviar e receber arquivo façam envio e recebimento byte a byte.