

Função generate_random_password:

Gera uma senha aleatória composta por letras maiúsculas, minúsculas e dígitos, com um comprimento de 8 caracteres.

Função string_arquivos:

Lista todos os arquivos no diretório compartilhado do cliente.

Calcula o hash MD5 de cada arquivo.

Retorna uma string contendo informações sobre cada arquivo no formato "hash, nome_arquivo".

Função find_port:

Cria um socket temporário.

Obtém uma porta disponível para o socket.

Fecha o socket temporário e retorna o número da porta.

Função calcular_hash_do_arquivo:

Calcula o hash de um arquivo dado o caminho, usando um algoritmo específico (padrão: SHA-256).

Utiliza um buffer para processar o arquivo em blocos.

Função listar_arquivos_e_calcular_hash:

Lista os arquivos no diretório especificado.

Para cada arquivo, calcula o hash e exibe uma lista numerada para seleção.

Função envia_recebe_udp:

Envia uma mensagem UDP para um servidor e recebe a resposta.

Utiliza um socket UDP para comunicação.

Função menu_selecionar_arquivo:

Apresenta ao usuário uma lista de arquivos disponíveis.

O usuário pode escolher um arquivo pelo número associado.

Retorna as informações do arquivo selecionado.

Função menu_selecionar_host:

Recebe informações sobre um arquivo (hash, nome, hosts).

Apresenta ao usuário uma lista de hosts associados ao arquivo.

Retorna o IP, porta, hash e nome do arquivo escolhido.

Função requisita_arquivo:

Estabelece uma conexão TCP com um host especificado.

Envia uma solicitação para obter um arquivo com base no hash.

Recebe e salva o arquivo no diretório compartilhado do cliente.

Função controle_udp:

Inicia um serviço de controle via UDP.

Oferece comandos para listar arquivos, baixar arquivos, atualizar registros e desconectar do servidor.

Função `servico_tcp`:

Lida com solicitações TCP para obter arquivos.

Recebe mensagens do cliente e fornece o arquivo correspondente, se existir.

Função `controle_tcp`:

Inicia um serviço TCP para lidar com solicitações de arquivos.

Aceita conexões de clientes e inicia threads para atendê-los.

Função `inicia_controle_tcp`:

Inicia o serviço de controle TCP.

Função `inicia_controle_udp`:

Inicia o serviço de controle UDP.

Função `find_available_tcp_port`:

Encontra uma porta TCP disponível usando um socket temporário.

Função `register_to_server`:

Registra o cliente no servidor com informações como senha, porta TCP e diretório compartilhado.

Função `configurar_ambiente`:

Configura o ambiente do cliente, gerando uma senha aleatória e encontrando uma porta TCP disponível.

Função `main`:

Configura o ambiente.

Inicia os serviços TCP e UDP em threads separadas.

Entra em um loop de espera.

Código principal: Lê argumentos da linha de comando para obter o IP do servidor e o diretório compartilhado. Chama a função `main` para iniciar a execução do cliente.

---- SERVIDOR ----**Função `calculate_md5`:**

Calcula e retorna o hash MD5 de um conteúdo de arquivo.

Função `handle_reg`:

Lida com mensagens de registro (REG).

Extraí senha, porta e arquivos do cliente da mensagem.

Registra o cliente no servidor e retorna uma resposta.

Função `handle_upd`:

Lida com mensagens de atualização (UPD).

Atualiza a lista de arquivos do cliente no servidor.

Retorna uma resposta indicando o sucesso da atualização.

Função `handle_lst`:

Lida com mensagens de listagem (LST).

Gera uma lista de todos os arquivos registrados pelos clientes.

Retorna a lista de arquivos como resposta.

Função `handle_end`: Lida com mensagens de encerramento (END). Remove o cliente do servidor e retorna uma resposta.

Função `main`: Inicia um servidor UDP. Fica aguardando mensagens dos clientes. Chama a função apropriada para lidar com cada tipo de mensagem. Envia respostas aos clientes.