

Relatório do Projeto de Redes de Computadores

Trabalho realizado por:

Marta Nunes, 2017246232

Pedro Nunes, 2016223060

Este projeto consiste em implementar um simples programa servidor que usa sockets TCP, que permite a cada utilizador deixar mensagens a outros utilizadores. Este programa pode ser facilmente testado com um dos programas cliente já existentes, como por exemplo o *netcat* ou *telnet*.

O código inicial é exatamente o mesmo apresentado na ficha 6 desta cadeira. Todas as funções são chamadas na função `'process_client()'`. Esta função tem um menu específico para o administrador e outro para os restantes utilizadores. Em cada um deles é pedido que se escolha uma opção do menu. Caso seja válida, essa opção e respetivas funções são executadas, e no fim disso o programa termina. Caso a opção seja inválida, a opção é pedida até que seja introduzida uma opção correta válida. Quando isso acontecer, ocorre o mesmo que se mencionou anteriormente para uma escolha válida.

Antes de começar a mencionar especificidades da parte do administrador e dos utilizadores, vejamos duas funções que foram introduzidas para facilitar a maneira como desenvolvemos o código: as funções `'ler_servidor()'` e `'replace()'`. A função `'replace()'` substitui um carácter por outro numa string C, e é de especial importância porque permite substituir o `'\n'`, correspondente à tecla 'enter' introduzida ao terminar a introdução de uma string C no *telnet* ou *netcat*, pelo carácter `'\0'`, o terminador de string. A função `'ler_servidor()'` lê o que é introduzido do lado do cliente e substitui o carácter `'\n'` pelo terminador de string. A declaração destas funções justifica-se pela regularidade com que são utilizadas, visto que caso contrário seriam uma repetição desnecessária de código.

Para gerir este servidor, temos o administrador chamado 'admin' que tem como função gerir os utilizadores guardados. Como tal, as únicas funções que pode desempenhar são inserir e remover utilizadores, sendo que não tem possibilidade de enviar mensagens a outros utilizadores.

Os utilizadores têm a opção de ver se existem mensagens enviadas por outros utilizadores, ler mensagens, enviar uma mensagem a outro utilizador, enviar a mesma mensagem a vários utilizadores e apagar mensagens lidas.

Quer seja um utilizador ou o administrador, a primeira função que é executada é a `'verificar_login()'`, que como o nome indica, verifica o login do cliente. Esta verificação é feita através do ficheiro `'users_passwords.txt'`. A função começa por pedir o nome de utilizador ao cliente e caso ele não exista no ficheiro devolve -1. No caso específico de não haver nenhum utilizador no ficheiro, o administrador tem que introduzir

manualmente o username e password no formato 'user|boss' em que o username é 'user' e a password é 'boss'. Caso o username introduzido exista, pede-lhe a sua password até que introduza a correta. Se o cliente for o administrador devolve 1 e se for um utilizador devolve 0. Estes valores devolvidos são utilizados na função 'process_client()' para chamar o menu e respetivas funcionalidades do utilizador, administrador ou terminar o programa diretamente.

Para o administrador, começemos por analisar a funcionalidade de adicionar novos utilizadores. Esta funcionalidade é cumprida pela função 'adiciona_user()'. Esta função pede ao cliente (admin) o nome do utilizador a adicionar, assim como a sua password e escreve-os em 'users_passwords.txt'. Para além disso abre um ficheiro de texto que corresponde à caixa de entrada específica a este novo utilizador. Caso não haja problemas nesta operação o cliente (admin) recebe uma mensagem de confirmação de sucesso nesta operação.

Para além desta funcionalidade, o cliente (admin) também pode remover um utilizador, na função 'remove_user()'. Para tal, pede-se o nome que o cliente (admin) quer remover. Se for introduzido 'admin', o programa termina com a mensagem 'Não pode eliminar o admin.'. Caso contrário, começa a procurar no ficheiro 'user_passwords.txt' pelo utilizador que se pretende remover. Se não existir nenhum utilizador com o nome introduzido, o programa termina com a mensagem 'Não existe nenhum utilizador com esse nome.' e se existir, gera o nome da caixa de entrada do utilizador e apaga-a. No fim termina com uma mensagem de sucesso. Este processo de apagar o utilizador e respetiva password é feito na função 'apagar_linha()' da seguinte forma: percorre todas as linhas do ficheiro original e apenas copia para um ficheiro temporário os usernames e passwords que não são iguais ao que se pretende remover. Por fim, remove-se o ficheiro original e põe-se o seu nome no ficheiro temporário, que nesta altura deixa de ser temporário.

Passemos agora à parte dos utilizadores. A primeira funcionalidade é ver se existe alguma mensagem na caixa de entrada do utilizador. Visto que o login já foi feito anteriormente, agora só necessitamos de abrir o ficheiro correspondente à sua caixa de entrada. Para tal, se não houver erros na abertura do ficheiro, apenas se verifica se o ficheiro está vazio ou não, e envia-se uma mensagem para o cliente (utilizador), a dizer se a caixa de entrada está vazia ou não.

Em seguida, temos a funcionalidade de ler todas as mensagens, que é concretizada com a função 'ler_mensagens()'. Para tal, basta ler todas as mensagens e utilizador que as enviou, que estão no ficheiro de caixa de entrada e guardar numa variável para mostrar ao utilizador. Para além disso é necessário passar a flag que indica que a mensagem está lida para '1' para cada linha da caixa de entrada do utilizador (cliente).

Depois temos a função 'enviar_mensagem_a_um_utilizador()', que permite enviar uma mensagem uma mensagem a um outro utilizador. Para tal, é necessário pedir o nome do destinatário da mensagem. Se o utilizador introduzido for o próprio, ou o 'admin' a execução da função acaba com uma mensagem adequada. Se o utilizador

introduzido não existir acontece o mesmo. Caso o utilizador exista, após procura no ficheiro 'users_passwords.txt', é pedida a mensagem a enviar e enviada para a caixa de texto do destinatário no formato 'flag_lida|quem_enviou|mensagem'.

Temos agora a função 'enviar_mensagem_a_varios_utilizadores()', que é muito semelhante à anterior. A principal diferença é que a mensagem a enviar é pedida primeiro e depois é pedido o nome de cada destinatário em separado de modo a poder cada caixa de entrada em separado. As mesmas verificações de enviar para si próprio ou para o administrador são aplicáveis.

Por fim temos a função 'apagar_mensagens_lidas()', que usa novamente a abordagem de um ficheiro temporário, neste caso para a caixa de entrada, para apagar todas as mensagens lidas, ou seja, as que têm a flag de leitura a 1, no ficheiro da caixa de entrada do utilizador.