

1. Objetivo

O objetivo deste projeto é oferecer a experiência de desenvolver aplicativos de rede simultâneos. Para tal, é necessário utilizar a API do Berkeley Sockets para implementar um servidor de bate-papo da Internet usando um subconjunto do IRC (Internet Relay Chat Protocol RFC: <http://www.irchelp.org/irchelp/rfc/>). O IRC é um sistema global de bate-papo distribuído em tempo real que opera pela Internet. Uma rede de IRC consiste em um conjunto de servidores interconectados. Uma vez que os usuários estejam conectados a um servidor de IRC, eles podem conversar com outros usuários conectados a qualquer servidor na rede de IRC. O IRC fornece comunicação de grupo, através de canais nomeados, bem como comunicação pessoal através de mensagens “privadas”. Mais informações sobre o IRC, incluindo software de clientes e redes públicas de IRC podem ser consultados em <http://www.irchelp.org/irchelp/new2irc.html>. Neste projeto será implementado um ambiente simplificado do IRC, com um único servidor de IRC independente. Deve ser assumido que existe apenas um servidor de IRC e todos os clientes estão conectados a esse servidor.

2. Descrição das Funcionalidades do Servidor e do Cliente

O projeto deverá implementar um servidor IRC simplificado. O protocolo original do IRC é definido na RFC 1459. No projeto deverá ser desenvolvido um subconjunto do protocolo que tem como objetivo ilustrar o desenvolvimento de aplicativos de rede simultâneos. Especificamente, seu servidor deve implementar os seguintes comandos:

Comandos Básicos

- NICK - Dê ao usuário um apelido ou altere o anterior. Seu servidor deve informar uma mensagem de erro se um usuário tentar usar um apelido já recebido.
- USUARIO - Especifique o nome de usuário, *hostname* e nome real de um usuário.
- SAIR - Finaliza a sessão do cliente. O servidor deve anunciar a saída do cliente para todos os outros usuários que compartilham o canal com o cliente que parte.

Comandos de Canal

- ENTRAR - Comece a ouvir um canal específico. Embora o protocolo padrão de IRC permita que um cliente entre em múltiplos canais simultaneamente, seu servidor deve restringir um cliente a ser um membro de no máximo um canal. Ingressar em um novo canal deve implicitamente fazer com que o cliente saia do canal atual.
- SAIRC - Saída de um canal específico. Embora um usuário possa estar em apenas um canal por vez, SAIRC ainda deve manipular vários argumentos. Se nenhum canal desse tipo existir ou existir, mas o usuário não estiver nesse canal, envie a mensagem de erro apropriada.
- LISTAR - Lista todos os canais existentes apenas no servidor local. Seu servidor deve ignorar parâmetros e listar todos os canais e o número de usuários no servidor local em cada canal.

O servidor deve ser capaz de suportar vários clientes simultaneamente. O único limite para o número de clientes simultâneos deve ser o número de descritores de arquivos disponíveis no sistema operacional. Enquanto o servidor está esperando por um cliente para enviar o próximo comando, ele deve ser capaz de manipular entradas de outros clientes. Além disso, seu servidor não deve desligar se um cliente enviar apenas um comando parcial. O servidor deve ter suporte para conexões simultâneas, entretanto, o uso de *threads* não é permitido no projeto. Como teoricamente o IRC é um servidor público, sua implementação deve ser robusta e com tratamento de erros (por exemplo, caso seja usado um comando não válido) e o mesmo deve possibilitar aos clientes o seu acesso pelo nome do servidor ou pelo endereço IP. Espera-se que o servidor seja capaz de manipular vários comandos em um pacote. Entretanto, o servidor não deve estourar nenhum buffer quando o cliente envia uma mensagem que é muito longa (mais de 512 bytes). Em geral, seu servidor não deve estar vulnerável a um cliente malicioso.

O servidor pode ser desenvolvido em qualquer linguagem. Entretanto, não podem ser usadas bibliotecas de classes de *sockets* personalizadas, apenas a biblioteca de *sockets* padrão no sistema operacional Linux, dado que este projeto tem objetivos didático. Da mesma forma, não podem ser usadas bibliotecas para *threading*.

3. Detalhes do Relatório

O relatório do projeto deve ser entregue no formato habitual e o código desenvolvido deve constar como anexo. O código deve estar amplamente documentado. Na seção de Análise Experimental do relatório devem ser descritos e justificados os parâmetros de configuração do servidor IRC (nome, porta, etc.), devem ser mostrados e descritos exemplos de uso e com o uso do Wireshark deve ser analisado o fluxo de mensagens trocados entre um cliente e o servidor IRC, que devem ser coerentes com o código implementado. Nesta seção também deve ser incluído um *link* para um vídeo que faça uma demonstração (máximo 02 minutos) sobre o uso do servidor por todos os membros do grupo.