

Chat Cliente-Servidor IRC

Lucas Tourinho Mamede;

Mario Cesar Nobrega da Nobrega;

Marco Barem;

Gabriella Pereira;

Resumo

Este artigo apresenta a implementação de um servidor e cliente IRC (Internet Relay Chat) em Python. O objetivo é criar uma plataforma de comunicação em tempo real que permite aos usuários conectar-se a um servidor IRC, ingressar em canais de chat, trocar mensagens privadas e de grupo, e executar diversos comandos do protocolo IRC. A implementação foca em funcionalidades básicas e conformidade com o protocolo IRC. A utilização de sockets e threads permitiu gerenciar múltiplas conexões simultâneas, assegurando a responsividade e eficiência do sistema.

Arquitetura do Sistema

O sistema é composto por dois componentes principais: o servidor IRC e o cliente IRC. O servidor gerencia as conexões dos clientes, os canais de chat e os comandos IRC, enquanto o cliente permite que os usuários se conectem ao servidor e interajam com ele através de comandos e mensagens.

Tecnologias Utilizadas

Python 3.10: A linguagem de programação utilizada para implementar tanto o servidor quanto o cliente IRC. Python é escolhido por sua simplicidade e robustez na manipulação de sockets e threads.

Sockets: Utilizados para estabelecer a comunicação entre o servidor e os clientes, permitindo a troca de mensagens e comandos em tempo real.

Threads: Empregadas para gerenciar múltiplas conexões simultâneas, assegurando que o servidor possa atender vários clientes ao mesmo tempo sem bloqueios.

Descrição Geral do Funcionamento

O servidor IRC escuta em uma porta específica (por padrão, a porta 6667) e aceita conexões de clientes. Cada cliente é gerenciado por uma thread dedicada que lida com a recepção e processamento de mensagens. Os clientes podem enviar comandos IRC para o servidor, como NICK para definir um apelido, JOIN para ingressar em um canal, PRIVMSG para enviar mensagens privadas, entre outros. O cliente IRC permite ao usuário conectar-se ao servidor, definir seu apelido e nome real, ingressar em canais de chat, e enviar mensagens. O cliente também trata automaticamente respostas do servidor, como mensagens PING, respondendo com PONG para manter a conexão ativa.

Implementação do Servidor IRC

O servidor IRC é responsável por gerenciar as conexões dos clientes, os canais de chat, e os comandos IRC. A implementação utiliza a classe IRCServer para configurar o servidor e a classe ClientHandler para tratar a comunicação individual com cada cliente.

Gerenciamento de Conexões

O servidor é implementado utilizando sockets e threads. O socket do servidor escuta em uma porta específica, e cada nova conexão é aceita e gerenciada por uma instância da classe ClientHandler, que é executada em uma thread separada. Isso permite que o servidor atenda múltiplos clientes simultaneamente.

Tratamento de Comandos IRC

A classe ClientHandler implementa métodos para tratar os diferentes comandos IRC recebidos dos clientes. Cada comando é processado de acordo com as especificações do protocolo IRC. Por exemplo, o comando NICK define o apelido do cliente, enquanto o comando JOIN permite que o cliente entre em um canal de chat. A conformidade com o

protocolo IRC é assegurada, respondendo adequadamente a comandos desconhecidos ou malformados.

Gerenciamento de Canais e Mensagens

Os canais de chat são gerenciados pelo servidor, permitindo que os clientes ingressem em canais existentes ou criem novos canais. Mensagens enviadas para um canal são retransmitidas para todos os clientes conectados a esse canal, possibilitando a comunicação em grupo. Mensagens privadas também são suportadas, permitindo a comunicação direta entre dois clientes.

Implementação do Cliente IRC

O cliente IRC é implementado na classe `ClienteIRC`, que permite ao usuário conectar-se ao servidor, enviar comandos e mensagens, e receber respostas do servidor.

Conexão ao Servidor

O cliente se conecta ao servidor utilizando sockets, estabelecendo uma comunicação persistente. Após a conexão, uma thread é iniciada para escutar mensagens recebidas do servidor, garantindo que o cliente possa reagir em tempo real a comandos e mensagens.

Envio e Recebimento de Mensagens

O cliente permite ao usuário enviar comandos IRC, como `NICK`, `USER`, `JOIN`, `PART`, `PRIVMSG`, e `QUIT`. Comandos são enviados ao servidor através do socket, e as respostas do servidor são tratadas pela thread dedicada. Mensagens do tipo `PING` são automaticamente respondidas com `PONG` para manter a conexão ativa.