***Construção de uma aplicação - Modelo IRC chat***

116572 Redes de Computadores

Profa. Priscila Solís Barreto

Autores:

Arthur

Departamento de Ciência da Computação

Universidade de Brasília

Brasília - DF, Brasil

line 4-e-mail address if desired

Enzo Zanetti Celentano (211026495)

Departamento de Ciência da Computação

Universidade de Brasília

Brasília - DF, Brasil

enzozc3@gmail.com

Paulo Fernando Vilarim de Almeida (211043351)

Departamento de Ciência da Computação

Universidade de Brasília

Brasília - DF, Brasil

paulofvilarimda@gmail.com

*Resumo – O presente artigo é um relatório elaborado a partir de um projeto de disciplina, pelo Departamento de Ciência da Computação, na disciplina de Redes de Computadores. Seu desenvolvimento baseou-se em elaborar uma aplicação capaz de implementar um servidor de bate-papo Internet, utilizando um subconjunto do protocolo Internet Relay Chat (IRC).*

*Palavras-chave - Internet, IRC, bate-papo, aplicação*

*Abstract - This article is a report developed from a discipline project, by the Computer Science Department, in the Computer Networks discipline. Its development was based on the elaboration of an application capable of implementing an Internet chat server, using a subset of the Internet Relay Chat (IRC) protocol.*

*Keywords – Internet, IRC, chat, application*

# Introdução

Aplicações do modelo cliente servidor estão presentes nas mais variadas esferas do nosso dia a dia, no que tange ao uso de aplicações de Internet. Neste contexto, um exemplo seria o Internet Relay Chat (IRC), um sistema de bate-papo em texto, constituído de clientes operando como usuários da aplicação, canais de comunicação para a efetiva troca de mensagens entre os usuários, e um servidor capaz de suportar e manipular tais operações.

Sob tal pespectiva, o presente artigo descreve a elaboração de uma aplicação baseada no protocolo do IRC, e está dividido em: fundamentação teórica, onde serão abordados os conceitos necessários para entendimento do trabalho realizado; análise do ambiente experimental e dos resultados, onde serão abordadas as descrições de hardware e software utilizadas, bem como o exame das atividades executadas no desenvolvimento da aplicação; por fim, as conclusões elucidadas a partir do artigo.

# Fundamentação Teórica

## Conceitos teóricos

A priori, a camada de aplicação é uma abstração da seção de protocolos de comunicação responsáveis pela troca de mensagens entre as aplicações. Eles atuam junto com os protocolos da camada de transporte e são responsáveis pela comuicação final entre as aplicações dos dispositivos.

A posteriori, o IRC é um protocolo que atua na camada de aplicação sob o modelo cliente servidor, ou seja, ele estabelece um bate-papo para um cliente, quando este se conecta a um servidor, com capacidade para qualquer número de clientes, o qual começa a escutá-los e a repassar suas mensagem para os demais usuários da aplicação. Outrossim, o servidor também deve ser capaz de interpretar requisições de comandos dos clientes, como a troca de seu apelido, listagem e locomoção entre os canais disponíveis, etc. As operações são feitas sob o protocolo de transporte TCP, devido à transferência de dados fim-a-fim ser realizada de forma ordenada, bidirecional e consideravelmente confiável, onde os pacotes transmitidos possuem um bit de reconhecimento, conhecido como Ack.

O IRC foi criado em 1998 e já passou por diversas mudanças desde sua primeira publicação, existindo mais de uma Request for Comments (RFC) para descrevê-lo. O IRC moderno possui uma infinidade de recursos, tais quais a presença de bots, para facilitar algumas operações do servidor, e a detecção de proxy, para arbitrar a conexão com um usuário detentor de um servidor proxy inseguro. Todavia, alguns padrões entre as implementações do IRC ainda se seguem, e dentre elas: a utilização da porta TCP 6667 ou semelhantes; a presença de canais de comunicação; a possibilidade do usuário registrar um apelido no bate-papo; a possibiliade do cliente enviar uma mensagem privada para um usuário específico, entre outras funcionalidades.

Ademais, pelo lado do cliente, tal usuário pode ser executado nos mais variados tipos de máquinas, tanto com diferentes hardwares, como um desktop ou um smartphone, quanto com diferentes softwares, como sistemas operacionais Windows ou Linux. Sob esse viés, o participante do bate-papo IRC estará presente na aplicação com certas atribuições, como um identificador – podendo ser uma máscara de host -, um apelido, um nome de usuário e o nome de seu host - o qual é recomendado ser borrado, para devida segurança e proteção dos clientes.

Ao se conectar com o servidor, o cliente IRC poderá lhe enviar pacotes de mensagens, enquanto ainda escuta e recebe outros pacotes do servidor IRC. Os pacotes que chegam no servidor serão examinados, devidamente manipulados e, então, serão entregues para os respectivos destinatários das mensagens, podendo ser transmitidas em um canal aberto ou de forma privada para um específico usuário.

## Técnicas utilizadas

Para o desenvolvimento da aplicação, foram utilizados sockets na linguagem de programação Python, o programa WIRESHARK, para validação da funcionalidade do código em conectar diferentes dispositivos na mesma aplicação e para exame dos pacotes trocados na rede, além da plataforma GitHub, para relato e documentação do código compartilhado entre os 3 autores do projeto.

# III. Ambiente Experimental e Análise de Resultados

## Descrição do Cenário

## Configurações utilizadas

Cada autor teve um ambiente de desenvolvimento diferente, com seus próprios dispositivos e respectivas configurações:

## *Arthur*

Escrever;

## *Enzo Zanetti Celentano*

Escrever;

## *Paulo Fernando Vilarim de Almeida*

A implementação e o experimento do código foram realizados por meio de um notebook com dois sistemas operacionais distintos, Windows 11 e Windows Subsystem for Linux (WSL) com distribuição Ubuntu, onde cada um possui uma repartição em disco distinta, além de diferentes endereços IPv4. Outrossim, o único ajuste necessário para execução da aplicação foi a instalação do python3 em ambas as máquinas.

## Linguagem de programação e bibliotecas

A linguagem de programação utilizada foi o Python, por motivações de domínio da sintaxe da linguagem por parte dos autores do projeto. Dessa forma, as seguintes bibliotecas foram utilizadas:

## socket

Para instanciamento de sockets na rede, capazes de estabelecer a comunicação entre os dispositivos alocados e adequadamente conectados;

## json

Para troca de mensagens realizada de forma persistente, onde os pacotes eram convertidos em json na transmissão pela rede;

## threading

Para realização de múltiplas tarefas e chamadas de métodos ocorrerem em fluxos de execução diferentes, possibilitando, assim, tanto o servidor quanto o cliente escutarem e enviarem pacotes simultaneamente;

## time

Para colocar os hosts em repouso em momentos adequados, ou seja, com necessidade de espera da execução de uma tarefa pelo outro lado da rede (cliente esperando servidor, ou servidor esperando cliente), a fim de se evitar conflitos e erros no envio e recebimento dos pacotes.

## *Configurações na rede*

A fim de se adequar às necessidades do protocolo IRC, o código foi implementado no modelo cliente-servidor, com a criação de duas classes de objetos, uma para o cliente, denominada Cliente, e outra para o servidor, denominada Servidor, onde cada uma se aloca em um arquivo “.py” distinto, e basta executar a interpretação do arquivo da classe que deseja instanciar um objeto e, enfim, rodá-lo em sua máquina. Em suma, a inteface do usuário é constituída pelo console, ou seja, por comandos de terminal realizados no repositório o qual armazena o código em execução.

Ademais, ao instanciar sockets do servidor e do cliente, foram utilizados os argumentos:

## socket.AF\_INET

A fim de se utilizar a família de endereços IPv4, ou seja, endereços IP constituídos por quatro grupos de dois dígitos hexadecimais, variando de 0 a 255 na base decimal em exibição;

## socket.SOCK\_STREAM

A fim de se utilizar uma transmissão de pacotes compostos por sequências de bits, sob o protocolo de transporte TCP, onde a delegação é realizada de forma ordenada, bidirecional e suficientemente confiável.

## Análise de Resultados

* (Atividade 1 realizada, descrevendo problemas encontrados e soluções tomadas).
* (Atividade 2 realizada, descrevendo problemas encontrados e soluções tomadas).
* ...

# IV. Conclusões

Em síntese, neste trabalho, percebe-se que existem diversas partes interconectadas as quais permitem o transporte de informações entre máquinas, pelo protocolo TCP. Não obstante, verifica-se também a necessidade da leitura e da compreensão de partes do protocolo para sua utilização neste projeto – especialmente, no que tange à criação e conexão dos sockets utilizados. Com isso, é crucial ressaltar que toda rede contemporânea usufrui dos assuntos abordados no trabalho, mesmo em linguagens de mais alto nível, em que as conexões são realizadas de maneira mais implícita.

##### Bibliografia

1. Site para consulta: https://pt.wikipedia.org/wiki/Internet\_Relay\_Chat
2. RFC para consulta: http://www.cs.cmu.edu/~srini/15-441/F06/project1/rfc.html
3. Exemplo de programação com socket, utilizado no começo do projeto: https://github.com/Gabrielcarvfer/Redes-de-Computadores-UnB
4. Bibliografia 4