



Universidade Federal de Pernambuco - UFPE
Centro de Informática – CIn
Curso de Bacharelado em Sistemas de Informação (BSI)



Lucas Florêncio de Sousa

lfs@cin.ufpe.br

PROJETOS DE DESENVOLVIMENTO COM SOCKETS

Orientação: Prof. Kevin Lopes e Prof. Ygor
Amaral
Disciplina IF975 - Redes de Computadores
2021.1

RECIFE, PE
2021

SUMÁRIO

- 1. INTRODUÇÃO
- 2. DESENVOLVIMENTO
 - 2.1 PROJETO 1
 - 2.1.1 Motivação
 - 2.1.2 Resumo
 - 2.1.3 Funcionamento
 - 2.1.4 Problemas/Possíveis Melhorias
 - 2.2 PROJETO 2
 - 2.2.1 Motivação
 - 2.2.2 Resumo
 - 2.2.3 Desenvolvimento
 - 2.2.4 Problemas/Possíveis Melhorias
- 3. CONCLUSÃO
- 4. REFERÊNCIAS

1. INTRODUÇÃO

Os dois projetos demonstrados neste relatório constituem o resultado da entrega final da disciplina Redes de Computadores, de 60 horas de carga horária de estudos, ministrada pelo Prof. Kevin Lopes Dias no curso de Sistemas de Informação do Centro de Informática (CIn) da Universidade Federal de Pernambuco.

Enquanto o primeiro trata da modelagem de um protocolo baseado em UDP para estabelecimento de um jogo no modelo Quiz, o segundo projeto trata da edição do protocolo HTTP/1.1 TCP para um binário de download de arquivos.

2. DESENVOLVIMENTO

2.1 Projeto 1 (Quiz UDP)

2.1.1 Motivação

O projeto trata da implementação de um jogo no modelo quiz de perguntas em rede, implementando a arquitetura cliente-servidor.

2.1.2 Resumo

Quando receber o comando ‘iniciar’ de qualquer cliente conectado, o servidor irá chamar a função que inicia o jogo, enviando as perguntas para todos os clientes, recebendo as respostas e atualizando a pontuação de cada cliente(jogador). Ao fim de 5 rodadas, o servidor envia o ranking com pontuações para todos os clientes anunciando o vencedor.

2.1.3 Funcionamento

O servidor é levantado, carrega as perguntas e respostas de ‘quiz.txt’ e aguarda a conexão dos clientes.

Obs: O arquivo ‘quiz.txt’ armazena as perguntas e respostas no modelo

Cada cliente que é levantado envia automaticamente uma mensagem ‘ola servidor’ para se comunicar com o servidor e ‘estabelecer uma conexão’.

O servidor aguarda essas mensagens e, caso o cliente não esteja registrado em ‘clients’, ele faz o registro do cliente através da função ‘new_client()’, guarda o cliente como chave num novo espaço no dicionário ‘clients’ e envia mensagem informando ao cliente que ele foi conectado.

Quando um dos clientes conectados envia a mensagem ‘iniciar’, o servidor chama a função ‘play_game’ que, por sua vez, irá zerar todos os pontos dos clientes, contar 10 segundos e chamar as 5 rodadas em sequência.

Cada rodada é iniciada pela função ‘new_round()’, que sorteia uma pergunta até encontrar uma ainda não utilizada no jogo, envia essa pergunta a todos os

clientes presentes, inicia o número de threads de acordo com o número de clientes conectados e aguarda as respostas por 10 segundos.

Passados os 10 segundos, o servidor calcula a pontuação de cada cliente baseado nas respostas e atualiza a pontuacao no dicionario clients, terminando por enviar a cada cliente o seu resultado na rodada e sua pontuação atualizada.

2.1.4 Problemas/Possíveis Melhorias

O desenvolvimento com threads foi a maior dificuldade enfrentada por mim neste projeto, de forma que passei um tempo considerável tentando implementar o servidor desta forma e, por fim, decidi tentar outra abordagem com ele enviando mensagens a todos os clientes igualmente por meio da função broadcast(). Entendi apenas no final que eu poderia usar as threads apenas para receber a resposta de cada cliente em separado e assim consegui fazer o jogo funcionar, mas ainda creio que há pontos a melhorar na utilização das threads.

O projeto, mesmo funcionando, precisaria de mais algum tratamento nas respostas e no funcionamento das threads para evitar bugs por conta da desconexão dos clientes em meio a partida e algumas outras situações corriqueiras de um jogo em rede.

Não consegui travar a rodada quando um participante acerta a pergunta, dessa forma, o jogo permite que todos os participantes enviem suas respostas até o fim do tempo proposto (10 segundos).

2.2 Projeto 2 (Servidor HTTP)

2.2.1 Motivação

Este projeto trata da implementação de um servidor TCP HTTP/1.1 com apenas o método GET, que será capaz de retornar arquivos dos mais diversos tipos e tamanhos.

2.2.2 Resumo

Não consegui avançar muito na implementação deste projeto uma vez que tive problemas com gerenciamento de tempo, iniciei o projeto muito tarde.

2.2.3 Desenvolvimento

Entendida a proposta, busquei implementar primeiramente um servidor TCP com o qual o cliente pode estabelecer uma conexão para troca de mensagens. A partir do funcionamento dessa primeira abordagem, procurei entender como faria o servidor retornar o arquivo a partir do header HTTP e como faria a edição do método GET para atender aos requisitos do projeto.

2.2.4 Problemas/Possíveis Melhorias

Tive problemas com gerenciamento de tempo durante quase todo o semestre pois adentrei um novo emprego logo no início do período com treinamento em horário de aula. Por conta disso, decidi focar na finalização do primeiro projeto e, com o esgotamento do tempo, não consegui avançar neste aqui.

Ainda assim, creio que entendi a proposta e o que precisaria ser feito para a conclusão do projeto e por isso planejo terminá-lo em breve para fins de portfólio.

3. Conclusão

Neste projeto, pude aprender como funciona a troca de dados em rede com sockets, e baseado no que vimos durante o semestre na disciplina, implementar os protocolos TCP e UDP, além de experimentar um pouco do funcionamento de um servidor HTTP.

Com o primeiro projeto, a experiência de codar com threads e enviar e receber dados em rede com sockets agregou bastante no meu conhecimento (antes nulo) sobre o assunto. Mesmo que ainda longe de dominar, creio que será de grande utilidade no decorrer do curso e até mesmo da carreira profissional.

Mesmo não tendo administrado meu tempo suficientemente bem para concluir ambos os projetos, consegui entender o contexto da proposta bem como o que poderia ser feito para o segundo projeto. Infelizmente faltou tempo e organização da minha parte para conciliar os projetos com outras atividades.

Com isso, concluo que os projetos agregaram bastante o conhecimento adquirido nas aulas, e, assim, cumpriu muito bem a proposta inerente de um projeto final de uma disciplina como essa que cursamos, agregando diversas ferramentas novas e sintetizando a prática do o que vimos em teoria. Pretendo aperfeiçoar ambos para usar em meu portfólio futuramente.

4. Referências

James F. Kurose | Kate Ross | Computer Networking - A top-down approach.
Livro da disciplina

[socket — Low-level networking interface — Python 3.10.1 documentation](#)

Acesso em Dez/21

[UDP - Client and Server example programs in Python | Pythontic.com](#)

Acesso em Nov/21

[GitHub - twslankard/python-chatroom: a simple UDP python example](#)

Acesso em Dez/21

[Building a basic HTTP Server from scratch in Python | Codementor](#)

Acesso em Dez/21

[Implementing HTTP from socket. Using TCP socket to implement HTTP... | by Umangshrestha | Geek Culture | Medium](#)

Acesso em Dez/21

[70 perguntas de conhecimentos gerais - \(todamateria.com.br\)](#)

Acesso em Dez/21