

MC833

Relatório - Projeto 1

TCP

Marcio Ivan de Oliveira Coelho Filho 173645

Thomas Jun Yamasaki 177677

*18 de Abril de 2018*

**1. Introdução**

Para este projeto, implementamos um servidor e um cliente *TCP* em *C*, integrado com um sistema de banco de dados *MYSQL*, com o intuito inicial de aplicar os conceitos de socket e comunicação e uma futura comparação com servidor *UDP*, simulando um sistema de consultas de dados de disciplinas de uma universidade. Analisamos as implementações feitas e o tempo de transmissão entre o servidor e o cliente.

**2. Descrição geral e Casos de uso**

O servidor *TCP* e o cliente *TCP* foram implementados em *C* com base no código disponibilizado pelo livro *Beej's Guide to Network Programming* e nos códigos vistos em sala de aula.

Basicamente, o servidor executa todos os passos básicos (criação de *socket*, *binding*, *listening*) até chegar no primeiro *loop* principal, em que são executadas as funções *accept()*. Neste ponto é criado um novo processo, usando se a função *fork()*, para que o servidor possa continuar esperando por novas conexões. Tal processo é responsável por mandar informações lidas de um banco, de acordo com o que for pedido pelo cliente. O servidor atende as consultas de um dado cliente, que faz as requisições por meio de determinadas opções possíveis denominadas de *opcodes*, até que ele decida sair do sistema para fazer login em outro tipo de usuário ou mesmo encerrar o programa. Após a saída de um usuário do sistema, o servidor fica em *standby* até que receba outro pedido de conexão. A conexão com o banco de dados é feita por usuário, ou seja, podem ser estabelecidas múltiplas conexões para o uso do banco.

Para que o cliente estabeleça a conexão com o servidor, ele precisa especificar apenas o seu *IP* (as portas estão setadas em 8000), podendo assim fazer login como professor ou aluno, e fazer requisições de leitura e escrita de dados, de acordo com o que for listado como operação possível, até que decida sair do programa.

**3. Armazenamento e Estrutura de Dados do Servidor**

Para o armazenamento de dados do nosso servidor, utilizamos o *MYSQL*, que é integrado com o servidor fazendo uso de bibliotecas prontas. O *MYSQL* guarda informações dos tipos de usuários e dos dados que podem ser consultados pelos usuários. Além de leitura de dados, o banco permite inserção de dados caso o usuário tenha autoridade para tal. A autenticação é feita pelo servidor, e dependendo do usuário que logar no programa, ele pode fazer determinadas consultas.

**4. Detalhes da implementação do servidor TCP**

O servidor TCP comunica-se com o cliente usando as funções *read()* e *write()*. Na primeira parte do código, antes das operações principais, em que são decididos pontos triviais tais como *login* ou *exit* do programa e tipo de usuário (aluno ou professor), as mensagens tem tamanho de 1 *byte* (strings representando inteiros 0, 1, 2, 3..). Utilizamos tal método para facilitar a entrada dada por um usuário, controlar o tipo da entrada, pois seriam aceitos apenas as opções listadas para o usuário, e para evitar situações não esperadas com o uso do *buffer*. Já para a parte de envio e leitura de dados, foi preciso o uso de loops tanto para leitura quanto para a escrita, pois os conteúdos transmitidos poderiam ser maiores que o tamanho do buffer.

**5. Gráficos com os tempos de comunicação e tempo total de consulta/atualização. Estes tempos podem ser medidos em um cenário LAN ou WAN. Análise dos resultados**

*Para cada tipo de operação, os tempos apresentados devem ser a média de pelo menos 30 execuções. Mostrar também o intervalo de confiança com nível de confiança de 95%.*

**6. Conclusão**

Conseguimos implementar o servidor e o cliente com sucesso, mesmo tendo vários problemas de leitura e envio de dados, que foram esclarecidos e resolvidos com êxito. Aplicamos os conceitos de socket vistos em aula, o que proporcionou uma consolidação de nosso conhecimento e um aprofundamento sobre o tema.

**7. Referências**

1. *http://beej.us/guide/bgnet/ (Beej's Guide to Network Programming)*
2. *http://www.ic.unicamp.br/~edmundo/MC833/mc833/SocketTCP1S2018\_01.pdf*
3. *http://www.csd.uoc.gr/~hy556/material/tutorials/cs556-3rd-tutorial.pdf*
4. *http://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html*