

## Diário de Estudos

18/08

Resolução do exercício de laboratório [FRC2021.1: Exercício UDP - requisito para o Projeto 1 da disciplina \(unb.br\)](#)

Consulta das referências das funções utilizadas nos códigos udpClient.c e udpServer.c, através, notavelmente, da página [struct sockaddr\\_in, struct in\\_addr \(ufri.br\)](#)

Consulta sobre portas [Lista de portas dos protocolos TCP e UDP – Wikipédia, a enciclopédia livre \(wikipedia.org\)](#)

19/08, 20/08 e 21/08

Leitura do primeiro capítulo do livro Redes de Computadores e a Internet: Uma Abordagem Top-Down de Jim Kurose

22/08 a 28/08

Leitura do capítulo 1 e do capítulo 3 do livro Redes de Computadores de Tanenbaum.

29/08

Pesquisa intensiva sobre socket, em específico, como utilizar cada função e quais comportamentos esperados.

Pesquisa intensiva sobre a fila utilizada a partir do msgget e suas funções derivadas, a saber a partir do link [IPC:Message Queues:<sys/msg.h> \(cf.ac.uk\)](#)

Pesquisa sobre a fila de mensagens POSIX, aka MQueue, através do link de documentação [mq\\_overview\(7\) - Linux manual page \(man7.org\)](#), bem como suas demais funções.

Pequeno estudo sobre errno e sua utilização sobre funções stubs, a saber: fopen, mq\_open, etc..

### Problemas encontrados durante a implementação do protocolo

**Problema:** Definir um padrão de mensagem a ser enviado pela fila interprocessos – entre a camada N e a camada N-1. Motivação: caso a camada N queira enviar um arquivo para o host B, como a camada N-1 saberá que se acabaram os chunks do arquivo vindos da camada N?

**Solução:** Adotou-se um cabeçalho de 4 bytes para dizer o comprimento útil da mensagem (da esquerda para a direita). Caso o comprimento informado seja 0, significa um End Of File. Nesse caso a camada de enlace finalizará o processo de envio ali, enviando uma última PDU sinalizando o EOF para a camada par. Já para a camada N, se esta ler uma mensagem de comprimento 0 vinda da fila, significa que nenhum outro chunk é esperado e que o arquivo pode ser fechado e salvo.