Relatório: C-Chat

Rafael Renó Corrêa

Junho de 2025

Introdução

A comunicação representa um dos pilares da vida social humana, onde o desenvolvimento da tecnologia propiciou formas de comunicação cada vez mais rápidas e eficientes. Nesse cenário, desde a invenção do telégrafo até a consolidação da *internet* por fibra-ótica, os aspectos do dia-a-dia estão cada vez mais voltados para as diferentes mídias sociais e ferramentas de comunicação pela rede de computadores, como Whatsapp, Messenger, Telegram, etc.

Entretanto, apesar da tendência atual de crescimento e acessibilidade dessas formas de comunicação, a maioria desses serviços de comunicação interpessoal por computador pertence a um grupo seleto de organizações que, muitas vezes, utilizam dos dados de usuário para instrumentalizar estratégias de distribuição de anúncios ou manipulação política. E é nesse contexto que a ferramenta C-Chat pode ser útil, uma vez que apresenta um modelo de um sistema de comunicação sem armazenamento de qualquer tipo de informação de usuário em longo prazo, através de uma ferramenta sofisticada e eficiente, implementada em C.

O C-Chat funciona com uma arquitetura simples no modelo cliente-servidor, onde uma máquina servidora principal orquestra a comunicação de múltiplos clientes de forma concorrente. Para isso, implementa multiprocessamento e *multithreading* com uma estrutura robusta: No processo pai da aplicação servidora, o programa estabelece novas conexões com máquinas clientes. Uma vez estabelecida, o código é herdado por um processo filho que manipula a comunicação de entrada e saída de dados com o cliente utilizando duas *threads* separadas. Enquanto isso, o processo pai se prepara para estabelecer novas conexões.

Nesse cenário, as tecnologias utilizadas foram o arquivo de cabeçalho da biblioteca padrão de uso geral na linguagem de programação C (stdlib.h) e o arquivo de cabeçalho que fornece acesso a funções da API do sistema operacional POSIX em C (unistd.h) – para manipulação de processos. Além disso, para a manipulação de *threads*, foram utilizadas diretivas definidas pela biblioteca OpenMP (omp.h). A comunicação entre processos no mesmo grupo foi sincronizada pela implementação das bibliotecas de semáforo do (sys/sem.h), espaço de memória compartilhado (sys/shm.h), comunicação entre processos (sys/ipc.h) e tratamento de sinais (signal.h). Por fim, todo tipo de atividade em camada de rede foi implementada utilizando comunicação orientada à conexão (IPv4 e TCP/IP) por soquetes de rede (sys/socket.h).

Além de outras bibliotecas utilizadas para modularização de tarefas e simplificação do código, tais como: stdio.h, arpa/inet.h, netinet/in.h, string.h, wait.h e errno.h.