**Laboratório 2**

**Construindo um simples servidor de arquivo**

*O objetivo desta experiência é o aprendizado dos princípios de programação em rede, bem como a discussão de um protocolo de comunicação, a nível de aplicação.*

A experiência será a implementação de um simples servidor de arquivos um pouco mais robusto que o software dado como exemplo (client.cpp e server.cpp), que **não** usa o HTTP. **Você** vai aprender a construir uma comunicação entre dois nós de rede projetando o seu próprio protocolo.

**NÃO faz sentido usar ChatGPT** ou similares para nada neste lab. As dificuldades que você encontrar, enfrentar e resolver vão ajudar no seu aprendizado dos princípios de protocolos de rede.

O sistema a ser desenvolvido deve seguir as seguintes definições:

1. Sua aplicação roda em rede utilizando soquetes em linguagem C.
2. Crie o protocolo correspondente para sua aplicação rodar. Não se apoie em outro protocolo de aplicação: sua aplicação deve rodar diretamente sobre o nível de transporte, especificamente o TCP.
3. A experiência pode ser realizada por no máximo dois alunos: um aluno implementa um lado da comunicação, outro implementa outro lado. Todas as porções de software terão que usar soquetes em linguagem C. Identifique no relatório qual aluno fez qual lado da comunicação.
4. Você deve projetar um protocolo com as seguintes características:

- O cliente envia uma mensagem de requisição **MyGet** (do tipo da Get do http, mas com menos campos de cabeçalho). A MyGet especifica a URL com o caminho e o arquivo desejado. O servidor retorna uma mensagem de OK com o arquivo desejado pelo cliente.

- O cliente pode a qualquer momento enviar uma outra mensagem **MyLastAccess** que requer que o servidor retorne o instante de tempo do acesso anterior a esta mensagem que foi realizado por este mesmo cliente. Se a primeira mensagem do cliente for MyLastAccess, a resposta do servidor será **Last Access**=Null.

Em outras palavras, este protocolo não é totalmente *stateless*, pois o servidor deve manter alguma informação sobre o andamento da comunicação.

1. O protocolo e a implementação sendo desenvolvidos devem ser o mais robusto possível, à prova de problemas de rede, de mensagens, de usuário. A robustez é um dos critérios de avaliação.
2. Use threads para quando chegar uma requisição de arquivo seu programa não ficar bloqueado atendendo um só usuário.
3. O foco principal desta experiência é a programação de rede, não a interface com o usuário que pode ser muito simples, sem recursos gráficos, etc.

1. O relatório deve ter duas seções:

**(1) A solução**

Explique o protocolo e as mensagens que definiram.

Coloque uma breve explicação de seu código, um link para acesso, e a forma de reproduzir o experimento. Qualquer informação relevante para avaliar seu código deve ser incluída aqui.

**(2) Testes**  
Testes realizados: Situações para as quais você testou o programa e conseguiu validar seu código.

Submeta no classroom o relatório final com os devidos links.