Segundo Projeto de Pesquisa

1. Título:

Criação de aplicações cliente/servidor utilizando protocolos TCP e UDP.

2. Objetivos do projeto e descrição Geral

O objetivo desse trabalho é fazer com que os alunos compreendam *(i)* aspectos inerentes à programação Cliente/Servidor com sockets TCP e UDP, para tratamento de conexões simultâneas e gerência de diálogo; e *(ii)* saibam construir aplicações que utilizem a API-Socket.

3. Dicas/Requisitos

- Todos os códigos entregues devem ter suas funções devidamente documentadas, a fim de facilitar a conferância das soluções apresentadas.
- É vedado ao aluno a utilização de códigos de terceiros para resolução das questões apresentadas nessa especificação. Além disso, é vedado o plágio das respostas de outros alunos, ou coletadas na Internet.
- Em relação ao ambiente de preparação da aplicação, os alunos devem utilizar ambeinte Linux, com compilador gcc e bibliotecas típicas (já inseridas nos códigos disponibilizados).
- Além dos códigos, os alunos podem incluir documentação adicional a fim de melhorar a qualidade da entrega.
- Além das entregas especificadas na seção 5, os alunos podem criar funcionalidades adicionais, a fim de melhorar a qualidade da entrega, as quais devem ser documentadas e justificadas, para que possam ser avaliadas.

4. Requisitos e regras para o trabalho

Este projeto deve ser feito obedecendo as seguintes regras:

- O trabalho pode ser feito por grupos de até 2 alunos ou individualmente. Em todas as entregas, as identificações dos alunos devem ser feitas, a fim de garantir a obtenção dos pontos relativos à entrega.
- A entrega deve ser feito em arquivo único, compactado (zipado), contendo as pastas relativas a cada uma das solicitações feitas na seção 5 desse documento de especificação. Dentro de cada diretório, importante ter: *(i)* o código fonte criado, sem erros e bem documentado, e *(ii)* um arquivo README, contendo explicação sobre a solução gerada e eventuais limitações do código.
- A nota é individual e será dependente da nota que o aluno tirar em avaliação específica, posterior à entrega do projeto. Nesse caso, o aluno deverá responder às questões que forem relativas ao trabalho, ou questões similares às que foram discutidas no contexto do projeto. Outros requisitos como cumprimento das datas, trabalho coordenado em grupo e inserção de novas funcionalidades no projeto também serão consideradas para emissão da nota final.
- A nota final do trabalho será obtida pela soma das notas de cada uma das questões apresentadas na seção seguinte. Além das solicitações feitas, o aluno pode fornecer soluções mais sofisticadas para alguma as questões, a fim de melhorar a nota final do projeto. No entanto, é importante que as modificações continuem atendendo o que foi solicitado e com nível de complexidade maior, para que possa ser considerado para pontuação adicional.

5. Entregas do projeto

Para esse projeto, devem ser entregues quatro blocos de código, relativos às questões a seguir. Para cada questão, gerar um arquivo zipado, contendo código gerado e documentado, que deve ser postado no Moodle.

- Crie uma aplicação UDP simples, cujo fluxo de diálogo entre Cliente e Servidor seja bidirecional. Ou seja, os dois lados (cliente e servidor) enviam e recebem mensagens de um para o outro.
- Crie uma aplicação TCP, para diálogo cliente/servidor, para diálogo bidirecional, cujo lado servidor permaneça com a conexão ativa até que o cliente informe ao servidor que não há mais nada a enviar; quando isso ocorrer, apenas o cliente encerra a conexão, mas o servidor deve continuar ativo, aguardando novas conexões.
- Crie um código TCP, no qual o servidor consiga atender mais de cliente ao mesmo tempo, para promover diálogo bidirecional (*half-duplex*, ou seja, enquanto um dos lados fala o outro escuta e vice-versa, numa ordem pré-definida), usando a *system call fork()*. Nos testes de validação, é importante que vários clientes possam dialogar com o servidor, mas não é necessário que clientes dialoguem entre si.
- O aluno deve pesquisar a *system call select()* e produzir novo código TCP, de modo a implementar um código TCP no qual os clientes se conectem ao servidor TCP e consigam dialogar entre si, num diálogo *full-duplex*, ou seja, os dois lados podem iniciar o diálogo, sem uma ordem pré-definida. Nesse caso, tudo o que um usuário digitar deve chegar aos pares conectados ao servidor TCP naquele momento. Obs.: Uma referência sobre essa *system call* pode ser encontrada em https://man7.org/linux/man-pages/man2/select.2.html.