Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
Curso: Tecnologia em Sistemas para Internet
Disciplina: Programação em Ambiente de Rede
Professor: Givanaldo Rocha de Souza

Lista de Exercícios - Sockets TCP / UDP e Threads

Entrega e Apresentação: 12 e 16 de abril (40 pontos)

1. Desenvolver a aplicação **Eco**: o cliente lê do teclado e envia para o servidor. O servidor recebe e reenvia para o cliente. O cliente imprime o que recebeu no formato **`<IP-servidor> mensagem`** e reinicia a leitura do teclado.
2. Desenvolver a aplicação **Ping**: o cliente envia uma requisição para um endereço IP fornecido como parâmetro e o servidor apenas responde **`Server <ip> alive`**.
3. Desenvolver uma operação de transferência de arquivos entre dois computadores usando socket TCP. O cliente informará o nome do arquivo que deseja receber. O servidor, em caso de existência do arquivo, iniciará a transferência. Caso contrário, informará ao cliente que não possui o arquivo em seu sistema.
4. Desenvolver a aplicação **Cara ou Coroa**: o usuário digita o seu palpite (cara ou coroa). Em seguida, são disparadas duas threads, uma imprime "Cara" e a outra imprime "Coroa". Cada uma das threads dorme por um tempo aleatório entre 0 e 30 milissegundos. Ao pressionar Enter, o programa principal deve finalizar as threads e verificar se a última palavra que foi mostrada corresponde ao palpite. Em caso positivo, informar que o usuário acertou. Caso contrário, informar que errou.
5. Desenvolver uma aplicação para simular a dinâmica de uma corrida de Fórmula 1. Serão duas threads em que uma representa o carro de Felipe Massa e a outra representa o carro de Lewis Hamilton. As threads possuem as seguintes características:
   - Cada thread possui um loop com 65 interações (as voltas da pista).
   - A cada volta, a thread dorme durante um tempo aleatório entre 0 e 1 segundo, representando o tempo gasto para percorrer a volta.

   O programa deve aguardar as duas threads terminarem, representando a linha de chegada, e então informar quem venceu a corrida (dica: pesquisar o método *join()*).
6. Deseja-se implementar um sistema eletrônico de votação a partir de uma aplicação servidora e de aplicações clientes que se conectam através de sockets de rede, com as seguintes características:
   - A aplicação servidora, ao ser iniciada, solicita que o usuário entre com o número de candidatos e os respectivos códigos dos candidatos via console.
   - A aplicação servidora passa a aceitar conexões TCP na porta 1500 e pacotes UDP na porta 2000.
   - Quando uma aplicação cliente é iniciada, esta abre uma conexão com o servidor, via porta TCP 1500, recebe do servidor uma lista com os códigos dos candidatos e fecha a conexão.
   - A aplicação cliente passa a receber do usuário, via console de texto, o código do candidato a ser votado. Quando isto acontece, a aplicação cliente envia para o servidor, via porta UDP 2000, um pacote contendo um número inteiro representando o código do candidato que foi votado.
   - A aplicação servidora, ao receber um novo pacote UDP com um voto, atualiza o resultado da votação e mostra no console.