

Disciplina: Redes de Computadores
Professor: Luiz A. Rodrigues

Data Entrega: 31/outubro/2019
Realização: individual ou em duplas.

Trabalho Prático – Jogo da Velha em Rede

O objetivo do trabalho é implementar um protocolo para o Jogo da Velha em rede utilizando uma arquitetura híbrida cliente/servidor e Peer-to-Peer. Inicialmente os jogadores devem se registrar em um servidor informando o seu *nome* e o *número da porta* em que estarão aguardando conexões. O endereço IP e porta do servidor são conhecidos e informados manualmente pelo jogador no início da execução do programa.

O servidor deve armazenar e disponibilizar uma lista de jogadores conectados contendo o nome, o endereço IP (o mesmo utilizado no momento da conexão com o servidor) e o número da porta informados pelo jogador. Um jogador deve contatar o seu oponente para iniciar o jogo utilizando as informações do servidor.

A linguagem de programação é livre e a interface de jogo também. No entanto, para padronizar a aplicação, foi definido um protocolo modo texto, que deve ser seguido por todas as implementações. Este protocolo possui as seguintes especificações:

a) Comunicação entre jogadores e servidor: serviço UDP

USER nome porta

Mensagem UDP enviada periodicamente (a cada 10 (dez) segundos) pelo jogador para o servidor indicando a sua presença no sistema. O nome é composto por uma única cadeia de caracteres e pode conter letras, números e _ (*underline*). O parâmetro *porta* indica em qual porta jogador aceitará conexões TCP de outros jogadores.

O sucesso do registro é confirmado pela mensagem **USER OK**. Tentativas de registros de jogadores com parâmetros inválidos são recusadas com a mensagem **USER NOK**. Cada *host* (endereço IP) pode registrar um único jogador. O registro de um novo jogador do mesmo *host* implica na substituição automática do anterior. Se o servidor não receber a mensagem de anúncio após 1 (um) minuto, o jogador é removido pelo servidor da lista de jogadores ativos.

LIST

Mensagem UDP utilizada pelo jogador para obter a lista atualizada de jogadores registrados no servidor. O retorno é **LIST N <jogador_1> <jogador_2> .. <jogador_n>**. O parâmetro *N* informa o número de jogadores da lista. Cada jogador é identificado por **nome:endereçoIP:porta**.

EXIT

Mensagem UDP enviada do jogador para o servidor indicando que o jogador está deixando o sistema. A ausência de mensagens **USER** após 1 (um) minuto tem o mesmo efeito.

b) Funções de jogo: serviço TCP

START nome

Mensagem TCP enviada por um jogador A para o jogador B quando A deseja se comunicar com B. O parâmetro **nome** refere-se ao jogador A. A resposta deve ser **START OK** para confirmar o início do jogo ou **BYE** para indicar a recusa.

BYE

Mensagem TCP enviada por um jogador A para o jogador B indicando que ele recusando o jogo (após uma mensagem **START**) ou deixando o jogo em andamento (a qualquer momento).

PLAY linha coluna

Mensagem TCP enviada pelo jogador para indicar a sua jogada. **linha** e **coluna** referem-se à posição no tabuleiro que o jogador deseja fazer a marcação, sendo $0 \leq \text{linha}, \text{coluna} < 3$. Uma jogada válida é confirmada com **PLAY OK**. Uma indicação de jogada incorreta (fora dos limites do tabuleiro ou em uma posição já marcada) deve ser recusada com a mensagem **PLAY NOK**, indicando que o jogador deve refazer a jogada. Três jogadas incorretas consecutivas implicam na anulação da partida com o envio de **BYE** pelo jogador que está aguardando a jogada.

Após cada jogada válida (com confirmação **PLAY OK** do oponente) o sistema verifica se há ganhador ou empate e informa ao jogador local.

A avaliação do trabalho será baseada em três partes:

- 1) Relatório contendo contextualização do problema, solução utilizada, protocolo utilizado, detalhes de implementação e resultados. O relatório deverá estar no formato de artigos da SBC (Sociedade Brasileira de Computação – <http://www.sbc.org.br/documentos-da-sbc/category/169-templates-para-artigos-e-capitulos-de-livros>).
- 2) Código fonte;
- 3) Apresentação conjunta no laboratório da execução do sistema.

Obs.: O relatório e o código devem ser entregues através do Moodle em um único arquivo no formato aluno1_aluno2.zip. Apenas um dos membros da equipe deve enviar o arquivo.