

MAC0448/5910 - Programação para Redes de Computadores

EP2

Data de Entrega: 16/10/2015

Prof. Daniel Macêdo Batista

1 Problema

Neste EP você deverá implementar um sistema de jogo da velha seguindo um protocolo definido por você. O sistema deve ser composto de dois códigos separados. Um código servidor, que centralizará as informações de login, status das salas dos jogos (quem está jogando com quem) e pontuação dos jogadores e um código cliente, que conectará no servidor e a partir de uma lista dos jogadores conectados permitirá a escolha do adversário. O servidor deverá ainda manter estado das partidas. Ou seja, se por algum motivo um cliente desconectar por algum problema na rede ou no sistema operacional do jogador¹, quando esse cliente voltar a conectar ele deve receber a informação de que a conexão anterior falhou e, se o jogador contra quem ele estava jogando ainda estiver ativo e sem jogar com outro jogador, ele receberá a opção de continuar a partida anterior do ponto onde ele estava.

O sistema deve funcionar tanto sobre TCP quanto sobre UDP e se for com TCP ele deve criptografar as informações de usuário e senha usando TLS.

Diferente do EP1, neste EP você deverá entregar, além dos códigos, um .pdf com slides para o caso de você ser sorteado para apresentação/arguição. Os slides deverão apresentar o seu protocolo, informando se ele foi baseado em algum protocolo existente ou se você criou ele do zero. Outras decisões de projeto que você julgar que merecem ser apresentadas também podem ser incluídas. Você terá no máximo 10 minutos para fazer sua apresentação caso seja sorteado, portanto cuidado para não preparar slides muito extensos.

Entregas sem o .pdf da apresentação não serão corrigidas e receberão nota ZERO.

2 Requisitos

2.1 Servidor

O seu servidor deve atender aos seguintes requisitos:

- Suportar conexões TCP e UDP com os clientes

¹O problema na rede será simulado na correção com a desconexão da placa de rede. O problema no sistema operacional será simulado na correção com um `kill -9` no processo do cliente

- Conversar com os clientes para:
 - login
 - verificação periódica de que os clientes continuam conectados (*heartbeat*)
 - logout
 - manutenção do estado das partidas
- No caso da conexão requisitada pelo cliente ser TCP, o envio das informações de usuário e senha no login com o servidor deve ser feito utilizando TLS
- Manter a lista de jogadores conectados
- Manter um arquivo de log informando tudo que aconteceu durante o tempo em que ele ficou ativo. Esse arquivo de log deve informar o momento do evento e qual foi o evento

2.2 Cliente

O seu cliente deve atender aos seguintes requisitos:

- Suportar conexões TCP e UDP com o servidor e com os outros clientes (um argumento na linha de comando da execução do cliente deve informar se a conexão deve ser feita com TCP ou UDP)
- Trocar mensagens de texto, como num bate-papo, com outros clientes diretamente sem passar pelo servidor
- Conversar com o servidor para:
 - login
 - informação periódica de que continua conectado (*heartbeat*)
 - logout
 - manutenção do estado das partidas
- No caso da conexão requisitada pelo cliente ser TCP, o envio das informações de usuário e senha no login com o servidor deve ser feito utilizando TLS

Se o cliente conectar com o servidor utilizando UDP ele só poderá conversar com outros clientes também usando UDP. Ou seja, a lista de possíveis jogadores com quem ele pode conversar deve ser filtrada para mostrar para ele apenas aqueles usando UDP. O mesmo vale para TCP.

2.3 Protocolo

Você precisa criar um protocolo para permitir a comunicação entre clientes e servidor. Esse protocolo deve usar comandos em ASCII para pelo menos as seguintes ações:

- cadastro de novo usuário
- login
- logout

- *heartbeat* entre cliente e servidor
- envio e recepção de mensagem de texto entre clientes
- listar jogadores conectados no mesmo protocolo (TCP ou UDP), desde quando eles estão conectados, se estão ociosos ou se estão jogando com algum outro jogador
- solicitar início de partida com um dado jogador
- solicitar encerramento de partida com o jogador corrente
- informar onde o X ou o O foi colocado no tabuleiro
- informar o *Hall of Fame* com a pontuação de cada um dos jogadores mantida atualizada (se o servidor for desligado, ele deve recuperar essa informação quando for reiniciado)

O protocolo e os códigos devem considerar que um jogador vai entrar no sistema e vai jogar no máximo com um outro jogador por vez. A troca de mensagens pode ser realizada apenas com um jogador com o qual se está jogando e não deve ser feita usando o servidor como intermediário. Já a colocação de X e O no tabuleiro deve ser feita usando o servidor como intermediário. O servidor, ao receber o X ou o O de um jogador deve enviar essa informação para o outro jogador ver o tabuleiro atualizado. Também é papel do servidor verificar se a partida terminou por empate ou vitória. Para computar a pontuação de cada jogador considere que vitórias dão 2 pontos para o vencedor e empates dão 1 ponto para cada um dos dois jogadores.

O código do servidor não deve permitir que um jogador logue duas vezes mesmo que seja na mesma máquina. Por exemplo, se um jogador A logado em um cliente na máquina com IP 1.1.1.1 quiser logar em um cliente da máquina com IP 2.2.2.2 ele precisa primeiro rodar o logout no cliente que está em execução no IP 1.1.1.1.

2.4 Linguagem

Os programas podem ser escritos em qualquer linguagem de programação, desde que exista compilador gratuito para GNU/Linux, e devem funcionar no shell, sem interface gráfica. Certifique-se de que seu programa funciona no GNU/Linux pois ele será compilado e avaliado apenas neste sistema operacional.

Você não pode utilizar bibliotecas, classes ou similares que já implementem um sistema de jogo da velha. Códigos que não respeitem esse requisito terão nota ZERO.

2.5 Slides

Os seus slides devem ser feitos de modo a serem apresentados em um tempo máximo de 10 minutos. Os slides devem conter:

- Explicação do protocolo com destaque para informações sobre como foi feita a manutenção do estado das partidas
- Comentários sobre os testes do sistema quando houve desconexão da rede e falha do sistema operacional
- Conclusões focando nas dificuldades encontradas durante o desenvolvimento do EP e nos pontos positivos de ter realizado o EP, caso você ache que teve algum

3 Entrega

Você deverá entregar um arquivo .tar.gz contendo os seguintes itens:

- fonte do cliente e do servidor;
- Makefile (ou similar);
- arquivo LEIAME;
- .pdf dos slides.

O desempacotamento do arquivo .tar.gz deve produzir um diretório contendo os itens. O nome do diretório deve ser ep2-membros_da_equipe. Por exemplo: ep2-joao-maria.

A entrega do .tar.gz deve ser feita através do PACA.

O EP pode ser feito individualmente ou em dupla.

Obs.: Serão descontados pontos de EPs que não estejam nomeados como solicitado, que não criem o diretório com o nome correto após serem descompactados ou que não contenham todos os arquivos necessários.

Obs.: O prazo de entrega expira às 8:00:00 do dia 16/10/2015. Não serão aceitos EPs atrasados.

4 Avaliação

60% da nota será dada pela implementação, 10% pelo LEIAME e 30% pelos slides, independente deles serem apresentados. Os critérios detalhados da correção serão disponibilizados apenas quando as notas forem liberadas. Aqueles que forem sorteados para arguição e apresentação terão como nota final do EP a média entre a nota do .tar.gz (distribuída como explicado no início do parágrafo), a nota da arguição e a nota da apresentação.

5 Dicas

É altamente recomendável ler algum tutorial sobre programação com SSL/TLS na linguagem que você escolher. Se a linguagem for java o material em

<http://www.herongyang.com/JDK/SSL-Socket-Communication-Testing-Program.html> é suficiente. O básico sobre TLS também pode ser revisado na página da wikipedia em https://en.wikipedia.org/wiki/Transport_Layer_Security.