MAC0448/5910 - Programação para Redes de Computadores
EP2
Data de Entrega: 19/10/2014

Prof. Daniel Macêdo Batista

1 Problema

Neste EP você deverá implementar um sistema de bate-papo seguindo um protocolo definido por você. O sistema deve ser composto de dois códigos separados. Um código servidor que centralizará as informações de login e de tempos em tempos verificará se os clientes continuam conectados e um código cliente que deve se autenticar no servidor, informar para o servidor de tempos em tempos que está conectado, e se comunicar com outros clientes permitindo troca de mensagens e de arquivos. O sistema deve funcionar tanto sobre TCP quanto sobre UDP e se for com TCP deve criptografar a comunicação entre cliente e servidor usando TLS.

Você também deverá fazer experimentos para avaliar o desempenho do seu código em diversas situações. 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 e os resultados dos experimentos realizados, preferencialmente, em um ambiente virtualizado.

2 Requisitos

2.1 Servidor

O seu servidor deve atender aos seguintes requisitos:

- Suportar conexões TCP e UDP com os clientes
- Conversar com os clientes para:

  login

  verificação periódica de que os clientes continuam conectados (*heartbeat*)

  logout
- No caso da conexão requisitada pelo cliente ser TCP, a conexão com o servidor (**apenas com o servidor**) deve ser feita utilizando TLS
- Manter a lista de usuários 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
- Trocar mensagens e arquivos com outros clientes diretamente sem passar pelo servidor
- Conversar com o servidor para:

  login

  informação periódica de que continua conectado (*heartbeat*)

  logout
- No caso da conexão ser TCP, toda a comunicação com o servidor deve ser feita 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 usuários com quem ele pode conversar deve ser filtrada para mostrar para ele apenas aqueles usando UDP. O mesmo vale para TCP, lembrando que a comunicação direta com os clientes **não** deve usar TLS.

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:

- login
- logout
- envio e recepção de mensagem
- envio e recepção de arquivo
- *heartbeat* entre cliente e servidor
- listar usuários conectados no mesmo protocolo (TCP ou UDP) e desde quando eles estão conectados
- solicitar bate-papo com um dado usuário
- solicitar encerramento de bate-papo com o usuário corrente

O protocolo e os códigos devem considerar que um usuário vai conversar com no máximo um outro usuário por vez. Ou seja, se um usuário A estiver conversando com o usuário B e quiser iniciar a conversa com o usuário C, ele precisa primeiro encerrar a conversa com o usuário B. Entretanto nada pode impedir que dois códigos clientes sejam executados em uma mesma máquina. Por exemplo é possível que em uma mesma máquina haja n processos do código cliente em execução, cada um pertencente a um usuário diferente e se comunicando com outros usuários diferentes.

O código do servidor não deve permitir que um usuário logue duas vezes mesmo que seja na mesma máquina. Se um usuário 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 bate-papo. 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
- Detalhes do ambiente onde os experimentos foram realizados (por exemplo: configuração da máquina real e da máquina virtual, tanto em termos de hardware quanto em termos de software)
- Gráficos com resultados de experimentos conforme a lista mais abaixo
- Comentários sobre os experimentos (os resultados encontrados foram os esperados?)
- 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

Uma sugestão é que os experimentos sejam realizados em um computador com um máquina virtual com qualquer distribuição de Linux para facilitar a depuração do código. Qualquer software para virtualização pode ser utilizado mas fica a sugestão de utilizar o virtualbox (`https://www.virtualbox.org/`) pela facilidade de utilização. O servidor deve ser executado na máquina real, um cliente deve ser executado na máquina real e outro cliente deve ser executado na máquina virtual.

Os experimentos a serem realizados são os seguintes:

- Experimento 1: Medir o tempo para transferir arquivos de 10MB, 50MB e 100MB entre dois clientes. Para cada tamanho de arquivo você deve repetir o experimento 30 vezes e apresentar o resultado em gráficos em barra com a média e intervalo de confiança considerando 95% de confiança
- Experimento 2: Repetir o pedido no Experimento 1 mas agora utilizando UDP ao invés de TCP

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 ep1-membros_da_equipe. Por exemplo: ep1-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 19/10/2014. EPs entregues com atraso terão -1,0 por cada hora de atraso. Por exemplo, se você entregar seu EP entre 8:00:01 e 8:59:59, ele valerá 9,0

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`.

Se você não lembra de intervalo de confiança das aulas de estatística, leia sobre isso no livro do Raj Jain (`http://www.amazon.com/The-Computer-Systems-Performance-Analysis/dp/0471503363`). A Biblioteca do IME tem exemplares desse livro.