

Universidade Federal da Paraíba
Centro de Informática / Departamento de Informática
Disciplina: Redes de Computadores I – 3ª Avaliação Parcial

Orientações:

- São 60 alunos, distribuídos em 15 equipes com, no máximo, 4 alunos cada;
- Cada equipe deverá escolher entre uma das atividades a seguir (Projeto 1, Projeto 2 ou Projeto 3);
- Cada equipe deverá enviar email para giorgiamattos@gmail.com informando os integrantes da equipe e a atividade escolhida;
- **O que deve ser entregue:** os códigos fontes da implementação, relatório de como a atividade foi desenvolvida (dificuldades, facilidades, como a equipe resolveu/solucionou questões do projeto da atividade, e o que mais julgar necessário) e slides da apresentação, se houver, (70% da nota);
- Cada equipe deve apresentar, em 15 minutos no máximo, a sua atividade (30% da nota);
- As apresentações acontecerão nos dias **29/04, 06 e 08/05** e cada equipe deverá agendar a sua apresentação. Não deixe para o último dia!!

Observações:

Serão considerados na avaliação:

- Qualidade da implementação (robustez no tratamento de erros, legibilidade do código fonte);
- Corretude: se a aplicação roda e realmente faz o especificado para esta aplicação;
- Documentação;

Projeto 1: CHAT

Descrição:

Desenvolver uma aplicação que permita um chat, em grupo, para usuários em uma rede local. Não existe limites para o número de participantes do grupo. No chat, a qualquer momento um usuário participante poderá enviar/receber mensagens sendo que uma mensagem poderá ser enviada ao chat (permitindo que todos os participantes a vejam) ou, reservadamente, a um usuário específico que esteja online.

Cada mensagem será mostrada para cada usuário, no seguinte formato:

"<IP>:<PORTA>/~<nome_usuario> : <mensagem> <hora-data>"

Onde:

<IP> -Número do IP de onde originou-se a mensagem.

<PORTA>: Número da porta de onde originou-se a mensagem, do IP descrito acima.

<nome_usuario>: nome do usuário.

<mensagem>: mensagem recebida.

<hora-data>: hora e data da mensagem recebida, de acordo com o horário do servidor.

Deverão ser desenvolvidas duas aplicações seguindo a arquitetura cliente-servidor. A aplicação servidora gerencia os usuários conectados ao chat e as mensagens postadas nele. A aplicação cliente (usada por cada participante do grupo), em máquinas distintas.

Funcionalidades serão invocadas através de linhas de comando (não haverá menu) a serem interpretadas pelo servidor.

Funcionalidade	Comando	Restrição
Sair do grupo	bye	-
Enviar mensagem ao grupo	send -all <mensagem>	-
Enviar mensagem reservada	send -user <nome_usuario> <mensagem>	Notificar o usuário caso <nome_usuario> não exista
Visualizar participantes	list	-
Renomear usuário	rename <novo_nome>	Notificar o usuário: - Renomeado com sucesso ou; - Nome de usuário já em uso.

Observações:

- Se o usuário informar um comando diferente, o servidor deverá informar que tal comando é inválido;
- Toda a comunicação entre cliente-servidor deverá ser através do protocolo TCP;
- Existirá apenas um chat.

Projeto 2: JOGO DA VELHA

Descrição:

O projeto consiste em desenvolver um pequeno jogo com sockets UDP e TCP, usando a linguagem C, C++ ou Java. O jogo é para 2 jogadores onde as jogadas vão se alternando entre os jogadores. Cada grupo fará uma combinação de jogo+protocolo+arquitetura+linguagem e deve implementar o programa de acordo com os requisitos solicitados. Também deve produzir um texto discutindo como as questões a seguir foram abordadas.

Arquiteturas:

- cliente-servidor, direto - sem intermediários.
- clientes-central, clientes interagem através de uma central.

Cada arquitetura tem aspectos a serem considerados:

- Central: Onde vai ser executado? Como os clientes sabem onde está? Como se registrar? Como sincronizar o jogo e as jogadas?
- Direto: como?

Além do jogo, em si, o grupo terá que pensar em alguns pontos relevantes na aplicação:

- São dois jogadores que não se conhecem. Como um jogador obtém a referência do outro?
- Depois de obter a referência, como os dois jogadores tornam-se disponíveis para jogar, e combinam quem começa primeiro?
- Como as mensagens contendo as jogadas são representadas e manipuladas?
- Como terminar o jogo? Isto é: como terminar o jogo, decidir quem ganhou, notificar os jogadores e, finalmente liberar os recursos (socket, memória, etc.)?
- Um jogo pode ser interrompido no meio e depois reiniciado? E como ele recomeçará?
- O jogo possui requisitos de segurança contra trapaças? Por exemplo, um jogador consegue fazer duas jogadas seguidas antes do adversário fazer a sua?
- O que acontece se um jogador demorar muito na sua jogada?

Regras do Jogo da Velha

- O tabuleiro é uma matriz de três linhas por três colunas.
- Dois jogadores escolhem uma marcação cada um, geralmente um círculo (O) e um xis (X).

- Os jogadores jogam alternadamente, uma marcação por vez, numa lacuna que esteja vazia.
- O objetivo é conseguir três círculos ou três xis em linha, quer horizontal, vertical ou diagonal, e ao mesmo tempo, quando possível, impedir o adversário de ganhar na próxima jogada.
- Quando um jogador conquista o objetivo, costuma-se riscar os três símbolos.

Distribuição dos requisitos

Número	Grupo	Central	Direto	Java	C/C++	TCP	UDP	Jogo da Velha
1	Equipe 1	x		x	x	x		x
2	Equipe 2		x	x	x		x	x

Projeto 3: Sistema Tipo-Twitter

Descrição:

Desenvolver uma aplicação "tipo" o twitter onde haverá um único servidor que representará a pessoa a ser seguida. Nesse servidor deverão estar cadastradas as pessoas que o seguem.

Requisitos funcionais:

- seguir alguém (no caso, o servidor)
 - no servidor será mantida a lista dos nomes de seus seguidores, junto com seus endereços IP+Porta.
- deixar de seguir
 - o seguidor poderá enviar uma mensagem ao seguido, informando que não o seguirá mais (deixando de receber mensagens). Logo, o seguidor deverá ser removido da lista de seguidores existente no servidor.
- enviar mensagens aos seguidores
 - exibindo a data e hora que a mensagem foi enviada e o nome do servidor.
- enviar mensagem ao seguido
 - exibindo a data e hora que a mensagem foi enviada e por quem.
- tanto o servidor quanto os seguidores poderão, a qualquer momento, enviar mensagens e/ou recebê-las.
- toda mensagem postada por um seguidor deverá ser reencaminhada, pelo servidor, aos demais seguidores.
- o servidor deverá gerar um log (arquivo txt) contendo o histórico de todas as mensagens postadas. Cada linha do log deverá ter a tupla <nome do seguidor ou do seguido, IP+porta, data, hora, mensagem>.