



Universidade de Caxias do Sul
Centro de Ciências Exatas e Tecnologia – CCET
Disciplina: INF0203K –Sistemas Distribuídos
Período 2012/4
Professor: André Luis Martinotto

**1º Trabalho de Sistemas Distribuídos - Sockets
(5.0 Pontos - Em relação as notas de Trabalhos)**

Napster



Napster, criado por Shawn Fanning e seu co-fundador Sean Parker, foi um dos primeiros programa de compartilhamento de arquivos. O Napster mesclava centralização e descentralização. O servidor principal do Napster, hoje inexistente, mantinha listas contendo tanto os computadores conectados a ele quanto os arquivos compartilhados que essas máquinas tinham. Para se comunicar com o servidor principal, as máquinas deveriam ter o cliente local do Napster instalado.

Quando um cliente efetuava uma busca por arquivos na rede do Napster, o servidor respondia a partir da lista que mantinha dos clientes conectados. Caso fosse encontrado algo de interesse, o servidor tratava de mediar a comunicação entre o cliente que fez o pedido e o cliente que tem o arquivo. A partir daí, ambos os clientes se comunicavam por conexão direta, ou seja, os arquivos eram trocados diretamente entre pares, sem passar por servidores dedicados.

MyNapster - Descrição do Trabalho

Faça um servidor e um cliente para um sistema de compartilhamento de arquivos centralizado utilizando sockets. Esses devem apresentar uma forma de funcionamento semelhante ao Napster.

Descrição do cliente:

1. Ao se conectar ao servidor o cliente deve lhe enviar a lista com os nomes dos arquivos que possui no seu diretório compartilhado (diretório esse definido no cliente).
2. Quando o usuário fizer uma consulta ao cliente, esse deve:
 - a) Se conectar ao servidor e lhe enviar a solicitação e o servidor deve lhe retornar a lista dos arquivos cujo nome mais se aproximar do nome consultado, e o ip dos clientes conectados que possuem esse arquivo.
 - b) A lista de arquivos e IPs deve ser apresentada ao usuário, que escolherá qual arquivo ele fará o download, e de qual usuário.
 - c) O cliente deve tentar fazer conexão com o cliente que possui o arquivo e, se a conexão for feita, solicitar o arquivo.
3. O cliente também deve ficar esperando conexões de outros clientes e enviar os arquivos que forem solicitados.
4. O cliente também deve receber mensagens do servidor verificando se o cliente ainda está online.
5. Ao sair de uma sessão, o cliente deve enviar uma mensagem ao servidor que ira eliminar os índices do cliente de sua base.

Descrição do servidor

O servidor, por sua vez, deve:

1. Esperar conexões de clientes e receber deles a lista de arquivos disponíveis, e armazená-los, junto com o IP do cliente que os mantém.
2. Ao receber uma mensagem de desligamento, o servidor deve eliminar todos os índices daquele cliente do seu diretório.
3. Manter um controle dos clientes que estão online, e de tempos em tempos, verificar se ainda estão ativos.
4. Receber consultas de clientes e lhes retornar a lista de nomes de arquivos e IPs dos nomes mais próximos ao buscado.

Observações:

- Linguagens permitidas: Java, C#, C/C++, Python
- Grupos de no máximo três alunos
- Trabalhos iguais serão zerados.
- Data de entrega e apresentação: 18/05/2012.