IME - Instituto de Matemática e Estatística
DICC - Departamento de Informática
Sistemas Distribuídos / Tópicos Especiais

Prof. Alexandre Sztajnberg 2014/2

Lista 1 – Programação usando *Sockets*

Neste projeto vamos desenvolver um pequeno servidor de jogos com *sockets* TCP, no Unix (ou seja, "linha de comando"- não temos interface gráfica, a menos que você programe alguns Applets).

Crie um diretório L1 para conter os fontes e os executáveis. O professor vai avaliar os fontes e executar o jogo neste diretório.

Os jogos podem ser jogados por 2 jogadores, e são do tipo "turn" (as jogadas vão se alternado entre os jogadores). Exemplo: jogo da velha, batalha naval, xadrez, damas, maior-menor, par ou impar.

Podem jogar 2 jogadores externos, ou um jogador externo pode jogar contra o computador.

Arquitetura:

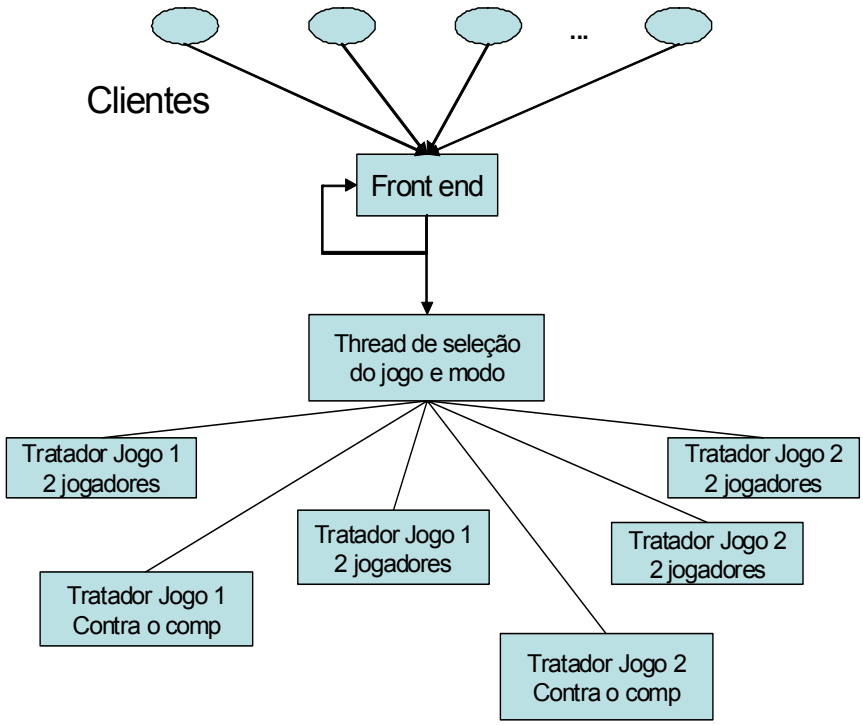

Figura 1. Arquitetura do sistema

Aspectos as serem considerados da arquitetura sugerida da Figura 1

- Frontend: onde vai ser executado?, como os clientes sabem onde está? Como registrar? Como sincronizar o jogo e as jogadas?

- No caso do cliente interagir com o computador, uma thread/processo tratador de conexão simples resolve
- Mas no caso de dois jogadores? Um tratador de conexão simples resolve? Provavelmente será necessário fazer o seguinte:
    - Primeiro jogador estabelece uma conexão e diz qual o jogo quer jogar e se quer um parceiro ou se vai jogar com o computador
    - Se quiser um parceiro, vai ter que aguardar um... ou ele já é o 2º jogador
    - O segundo jogador estabelece outra conexão com o FrontEnd. O *socket* ativo criado será passado para o tratador "anterior", que terá a referência a 2 sockets – um para cada jogador
    - Este tratador vai gerenciar o chaveamento dos turnos e repassará as jogadas de cada um, até o jogo terminar
- As jogadas devem ser representadas em XML. Para isso você vai ter que definir claramente o XSD (o template), com *tags* para os dois jogos para os dois jogos, etc.). Desta forma, cada jogada é, na verdade um grande *string* contendo XML. O Java tem toda a infraestrutura para fazer uso de XML. No C, será necessário usar alguma biblioteca. Podemos estudar alternativas.

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

- São dois jogadores que não se conhecem. Como um jogador obtém a referência do outro?
- Depois de obter a referência, como os dois jogadores "engage" no jogo? Isto é: tornam-se disponíveis para jogar, e combinam quem começa primeiro?
- Como as mensagens contendo as jogadas são representadas (a interface)?
- Como terminar o jogo? Isto é: como terminar o jogo, decidir quem ganhou, notificar os dois 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 "cheating"? 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?

Cada um deve também desenvolver um texto discutindo como as questões a seguir foram abordadas. O texto (só o texto, não os fontes e os executáveis) deve ser disponibilizado na conta através de uma página Web simples, mas organizada.