

Sistemas Operativos

Trabalho Prático - “CHAMPION”

Meta 3

João Pedro Verdete Santos	2017011382
Rafael de Jesus Saraiva	2017010339

Pormenores de Implementação

Árbitro

Na inicialização do árbitro o mesmo verifica se uma instância do árbitro já existe, para não existirem árbitros a duplicar. Após a inicialização fica à espera de comunicação por parte dos clientes ou por parte do administrador.

O árbitro tem a possibilidade de mudar no seu registo o jogo de um cliente (dentro da lista de jogos disponíveis), bem como desativar clientes individualmente e terminar a sessão atual (removendo ligações que tenha com os clientes avisando os mesmos).

Cliente

Na inicialização do cliente, é pedido o nome e confirmado com o servidor se o mesmo já existe, caso aconteça o cliente fica num loop até que seja introduzido um nome novo. Essa comunicação é feita por fifos.

É feita a distinção da terminação do cliente pela chegada ao fim do jogo (por saída do árbitro do campeonato, comando sair) ou pelo mesmo ter removido o utilizador do jogo (comando k<nome>).

Também é feita a distinção entre comandos internos (para o árbitro) e texto para o jogo. Sendo que o primeiro inicializam-se com #.

Principais Opções Tomadas

Árbitro

O armazenamento de jogadores é feito por uma lista ligada a uma estrutura.

A obtenção dos jogos para o campeonato, é feita automaticamente. Também guardada numa lista ligada de estrutura específica.

Usou-se um ponteiro do tipo diretoria aberta e funções de gestão de diretorias *opendir()* para obter a listagem de ficheiros na pasta GAMEDIR. Assim filtra-se os resultados e obtém-se os nomes dos jogos.

Decidiu-se usar um fifo principal para a comunicação geral (inscrições de clientes e comandos internos) e fifos secundários (um par por cliente) para a comunicação entre cliente e thread (que contém a instância do jogo associado).

Os fifos secundários são aos grupos de dois por cliente para tratarem do stdin e stdout do jogo, respetivamente.

Cliente

No que toca à interpretação de comandos na consola, o cliente faz uma simples filtragem do texto inserido para detectar se se trata de um comando interno ou de texto para o jogo. Para assim enviar essa informação pelo fifo apropriado.

Caso o cliente detecte uma mensagem externa, ele interpreta-a e age de maneira apropriada (por exemplo, no caso de “sair” espera por outra mensagem com a pontuação do cliente antes da saída do seu loop).

Funcionamento do programa

O árbitro consegue inserir comandos no terminal para obter informação sobre os jogos (games), jogadores (players), alterar o jogo do jogador (g[jogador]) e acabar as ligações a um cliente (k[jogador]) e a todos (exit).

O cliente consegue inserir os comandos #mygame para obter o nome do seu jogo atual e #quit para sair do campeonato. O resto do conteúdo inserido é considerado como input para o jogo.

Como foi feito

Cada cliente inscrito tem uma thread dedicada com dois fifos de comunicação . A informação do mesmo é guardada em lista ligada. A comunicação do jogo para o cliente é feita por fifos cujo output do jogo é redirecionado com dup's.

O que não foi feito

Timings do campeonato(detecção de número de jogadores para começar e duração máxima do campeonato).

Alterar o jogo funcional (só registo é alterado).

Possibilidade do cliente e árbitro reiniciarem sem reabrir os mesmos.

Suspensão e resumo de comunicação entre o jogador e jogo (com a devida informação).

O que não está a funcionar

Comunicação estável entre jogador e jogo.

Erros existentes

O jogo está sempre a assumir que há input do cliente levando à resposta constante no cliente, assumindo respostas erradas.