Relatório - Sistemas Operativos

João Pedro Caldas Cerqueira
a2020141650@isec.pt

Eduardo José Silva Bento
a2020139115@isec.pt

Servidor:

Estruturas de dados:
Game > contém toda a informação necessária para o jogo funcionar.
Vect2 > estrutura auxiliar para representar coordenadas.
SharedMem > contém a matriz do jogo e o buffer circular a recessão de mensagens.

O servidor é dividido em vários módulos:

sync.c > inicializa todos os handles e verifica erros.
init.c > inicializa todos os dados lógicos para que o programa inicie corretamente.
game.c > contém toda a lógica do jogo.
Server.c > inicializa todos os módulos apresentados anteriormente.

Monitor:

Estruturas de dados:
Game > contém toda a informação necessária para o monitor receber a informação do servidor.
Vect2 > estrutura auxiliar para representar coordenadas.
SharedMem > contém a matriz do jogo e o buffer circular a recessão de mensagens.

O monitor é dividido em vários módulos:

sync.c > abre todos os handles e verifica erros.
init.c > vai buscar a informação do registry.
game.c > imprime o mundo e envia informação para o servidor.
monitor.c > inicializa todos os módulos apresentados anteriormente.

Envio de informação do mundo para o monitor usando memória compartilhada.

É usada a função copyMemory() para copiar o estado atual do mundo para a estrutura de dados de memória compartilhada, é utilizado mutex para sincronismo apenas quando chamada essa função, após isso é enviado um evento que assinala do lado do monitor que deve imprimir novamente o mundo.

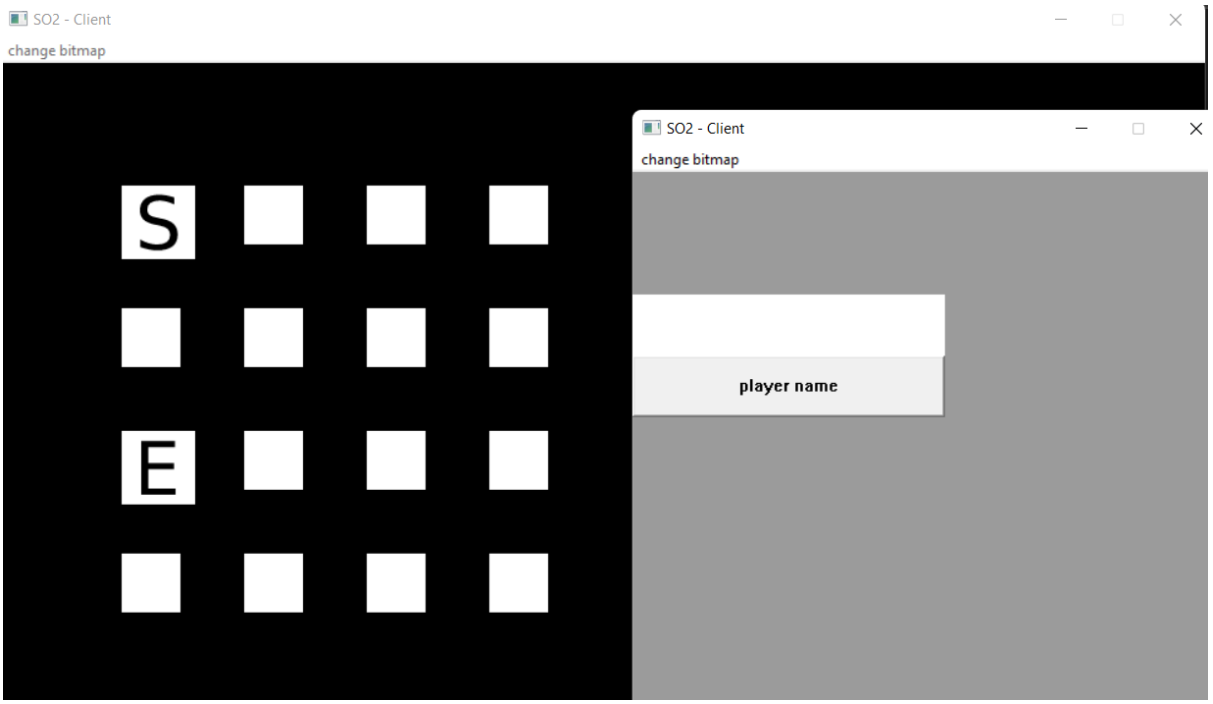
Envio de comandos para o servidor. (Buffer circular)

Utilizamos o modelo produtor-consumidor utilizado nas aulas a estrutura do buffer é um array de TCHAR.

Encerramento dos dois Programas

A estrutura Game apresenta um evento que ao fim do jogo é ativado (tem a flag de manual reset a TRUE). É utilizado o WaitForMultipleObjects() e após receber um dos eventos é avaliado se esse evento é o do fim do jogo e terminando a thread. A thread que contém o scanf para os comandos, ao fim de terminar a thread de imprimir o mundo é apresentada uma mensagem para pressionar alguma tecla para terminar o processo.

Cliente:



A comunicação entre o servidor é feita por named Pipes e sempre que o cliente quer enviar uma mensagem para o servidor é enviado uma evento para que do lado do servidor possa ser lido e vice-versa