

INSTITUTO SUPERIOR DE ENGENHARIA DE COIMBRA

Licenciatura de Engenharia Informática

Sistemas Operativos 2



Trabalho Prático – Especialíssima – Final

2016/2017

Fábio Coito

21240055

Engenharia Informática

Redes e Administração de Sistemas

Coimbra, novembro 2017

## Contextualização

Sou um aluno do ramo de redes e administração de sistemas. Neste momento tenho apenas a cadeira de Sistemas Operativos 2 por fazer para completar a licenciatura em Engenharia Informática.

Fui fazer exame em todas as épocas neste ano letivo sendo que na época normal desisti por não ter conhecimentos suficientes. Na época de recurso voltei a fazer exame e fiquei com a nota de 40%. Voltei ainda a tentar melhorar na época especialíssima, mas não consegui melhor.

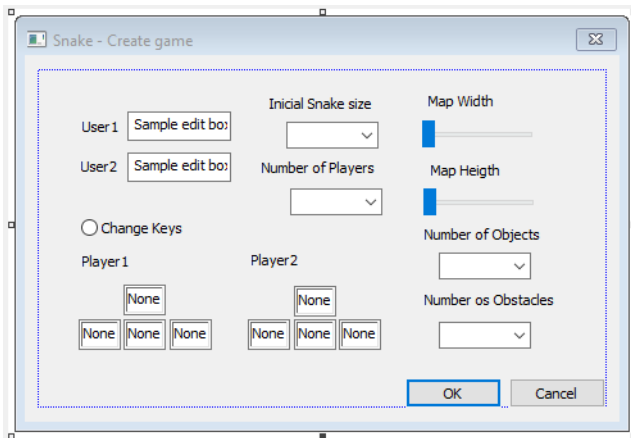
Quanto ao trabalho prático, entreguei na época normal. Estava muito incompleto sendo por isso que a nota que recebi foi de 19%. Na época especial não entreguei o trabalho porque tive um pouco ocupado com o relatório de estágio e outra cadeira que tinha pendente.

Quanto ao grupo, fiz sempre o trabalho sozinho sendo que os meus colegas do ramo de redes, todos desistiram da cadeira.

Pondo isto tenho em falta a entrega deste trabalho no qual preciso de cerca de 65% para ter nota final positiva e assim acabar o curso.

## Mecanismos do sistema

Para a comunicação entre o servidor e os clientes é usada a memória partilhada com várias vistas definidas para o conteúdo a memorizar. Para sincronizar os processos de leitura e escrita deveria ser usado um evento para cada ação. Ao iniciar o servidor devem ser criadas as estruturas necessárias ao jogo e colocadas na memória. Os clientes só podem iniciar com o servidor ligado sendo que a primeira coisa que fazem é associar a memória partilhada. Posteriormente devem ser indicadas as configurações para início de jogo. Nestas configurações deveria ser obrigatório apenas o nome de um jogador para iniciar. Podem ser escolhidas teclas para o movimento. O servidor deve receber a informação da tecla e converter para o sentido sendo por exemplo movimento para cima = 1.



## Estruturas de dados

As estruturas presentes são:

map – que contém informação sobre o tamanho, número e localização de objetos e obstáculos.

```
struct DLL_EXPORT map {
    int sizex = 30;
    int sizey = 30;
    int objects = 4;
    int obstacles = 4;
    int mapobject[MAX_MAP_OBJECTS][2]; // (x,y) coords
    int mapobstacles[MAX_MAP_OBSTACLES][2]; // (x,y) coords
};
```

snake – que contém o nome do jogador, tamanho, coordenadas e variáveis referentes a funcionalidades do jogo.

```
struct DLL_EXPORT snake {
    TCHAR username[MAX_TAM];
    int snakeSize = SNAKE_SIZE; // snake size
    int coords[MAX_SNAKE_SIZE][2]; // (x,y) coords
    int invertKeys; // invert input keys
    int oil; // raise speed
    int glue; // decrease speed
    int died; // 1 - died
    int points; // Current pontuation
    int currentDirection; // 1-Up 2-Down 3-Right 4-Left
};
```

game – é a estrutura que vai ser usada para manter os dados do jogo em memória e responsável pelo decorrer do jogo.

```
struct DLL_EXPORT game {
    snake snake[MAX_SNAKES];
    int numPlayers;
    int numBots;
};////
```

chat – é a estrutura que contem os ponteiros para os pipes referentes aos jogadores e também a mensagem enviada.

```
struct DLL_EXPORT chat {
    HANDLE owner;
    HANDLE players[MAX_SNAKES - 1];
};////
```

keys – é responsável por armazenar a informação no cliente correspondente as teclas escolhidas para o movimento

```
struct DLL_EXPORT keys {
    TCHAR username[MAX_TAM];
    TCHAR left;
    TCHAR righth;
    TCHAR bottom;
    TCHAR top;
};////
```

config – é responsável por armazenar as configurações do jogo na memória

```
struct config {
    TCHAR user1[MAX_TAM];
    TCHAR user2[MAX_TAM];
    BOOL changeKeys;
    keys kUser1;
    keys kUser2;
    int snakeSize;
    int numPlayers;
    int mapWith;
    int mapHeigt;
    int numObjects;
    int numObstacles;
};
```

change – estrutura responsável por passar a informação da tecla premida

```
struct change {
    TCHAR user[MAX_TAM];
    TCHAR userKey;
};
```

## Sumarização do ponto atual do trabalho

Devo dizer que o trabalho está bastante incompleto e não cumpre os requisitos. Demorei demasiado tempo a perceber a parte gráfica do trabalho e tentar estruturar o decorrer da execução. No trabalho que entreguei na época normal usei named pipes para a comunicação entre o servidor e os clientes. Para este queria usar memória partilhada o que ia facilitar o código necessário. O problema é que eu não percebi bem como usar a memória partilhada. Para o chat queria ter usado named pipes sendo que guardava na memória os handles e os clientes usavam esses handles para distribuir as mensagens. Contudo não percebi muito bem como isso ia ficar na parte gráfica sendo que penso que numa janela vai decorrer o jogo e com dois jogadores não faz muito sentido enviar mensagens para o colega.

Estive também a alterar todas as estruturas para facilitar a comunicação entre os clientes. Não testei ainda a criação de jogo e muito menos a impressão gráfica dos bmaps do jogo. Não tenho as estruturas para o uso dos bmaps. Esta ia ser a ultima parte que ia tentar fazer porque o que tinha em mente era conseguir que o servidor e o cliente comunicassem bem e que o chat funcionasse.

Posso concluir que devia ter me dedicado mais na realização deste trabalho até porque não percebo muito de programação e não sei estruturar um trabalho destes. Realmente achei que este é um trabalho para executar em mais de um mês. Era interessante ter concluído o trabalho porque poderia ser usado para alguma coisa. Espero que o próximo seja mais acessível para quem não queira ser programador.