

Relatório Trabalho Prático

Engenharia Informática



Trabalho Realizado Por:

- Beatriz Pinto 2020144095
- Eduardo Correia 2020139576

Índice

Conteúdo

1.Introdução	3
2.Linguagem Assembly	4
3.Estrutura do programa	5
4.Objetivos do programa.....	5
5.Funcionamento do programa.....	5
6.Algoritmo subjacente a cada função	6
7.Conclusão	7

1. Introdução

Este trabalho foi realizado no âmbito da Unidade Curricular de Tecnologias e Arquiteturas de Computadores como trabalho prático. Com esta atividade adquirimos e consolidamos conhecimentos sobre a linguagem de programação Assembly. Metemos em prática as coisas que aprendemos nas aulas e tentamos criar um jogo (jogo do labirinto).

2. Linguagem Assembly

Assembly é uma notação legível por humanos para o código de máquina que uma arquitetura de computador específica usa, utilizada para programar códigos entendidos por dispositivos computacionais, como microprocessadores e microcontroladores.

Enquanto um computador sabe o que a instrução-máquina *IA-32 (B0 61)* faz, para os programadores é mais fácil recordar a representação, equivalente, em instruções mnemónicas, como *MOV AL, 61h*. Esta instrução ordena que o valor hexadecimal 61 seja movido para o registo 'AL'.

A tradução do código *Assembly* para o código de máquina é feita pelo *assembler*. Ele converte os mnemónicos em códigos de operação, calcula os endereços de referências de memória e faz algumas outras operações para gerar o código de máquina que será executado pelo computador.

3. Estrutura do programa

O nosso programa é constituído por um menu que posteriormente redireciona o jogador ou para o jogo, composto por 5 níveis, com diferentes dificuldades, ou para o Top 10, que deveria apresentar as 10 melhores pontuações obtidas pelo jogador.

4. Objetivos do programa

Com este trabalho o nosso principal objetivo foi tentar aproximar ao máximo o nosso programa de um jogo jogável (jogo do labirinto) por um utilizador.

Para tornar este jogo o mais próximo do real possível, foi necessário criarmos funções que:

- impossibilitassem o jogador de atravessar as paredes;
- faça a contagem do tempo, de modo que o jogador não consiga ultrapassar o tempo máximo de cada nível;
- guarde as letras encontradas pelo jogador durante o jogo para completar as diferentes palavras de cada nível.

5. Funcionamento do programa

No início do nosso programa apresentamos um menu ao utilizador, que lhe permite escolher entre iniciar o jogo, o Top 10 (função não realizada) e ainda a opção de fechar o programa.

Se a opção do utilizador for iniciar o jogo, será redirecionado para o primeiro nível que apresentará uma palavra a ser encontrada pelo mesmo dentro do tempo previsto. Conforme o jogador complete as palavras de cada nível e não ultrapasse o limite do tempo, este irá ser enviado para o nível seguinte, em que a palavra e o tempo diferem. Tudo se repetirá até ao nível final (nível 5), se este nível for executado com sucesso o utilizador irá ser felicitado e encaminhado para o menu inicial. Caso algum dos níveis não seja efetuado com êxito, o jogador será alertado para a sua derrota, voltando também ao menu inicial.

6. Algoritmo subjacente a cada função

Ler_TEMPO: Esta função vai buscar as horas, minutos e segundos e guarda-as numa variável.

apaga_ecran: Função que apaga todo o conteúdo que aparece no ecrã.

IMP_FICH: Função utilizada para abrir o ficheiro que lhe é passado.

LE_TECLA: Esta função lê a tecla pressionada pelo utilizador e informa se se trata de uma tecla estendida ou de uma tecla “normal”.

Trata_Horas: Utilizamos esta função para imprimir a contagem do tempo de cada nível no monitor, e verificar se este foi ultrapassado.

AVATAR: Função principal do nosso programa que permite o funcionamento normal do jogo.

PAREDECIMA: Verifica se na posição acima do avatar existe uma parede, não deixando avançar nesse caso.

PAREDEBAIXO: Confirma se na posição abaixo do avatar existe uma parede, não deixando avançar nesse caso.

PAREDEDIREITA: Averigua se na posição à direita do avatar existe uma parede, não deixando avançar nesse caso.

PAREDEESQUERDA: Investiga se na posição à esquerda do avatar existe uma parede, não deixando avançar nesse caso.

Main: Função utilizada para chamar outras funções que permitem o bom funcionamento do jogo.

7. Conclusão

Concluindo com este trabalho conseguimos perceber mais sobre os registos do 8086. Através do desenvolvimento do jogo, conseguimos criar funções que evitassem que o avatar passa-se paredes, funções que contam o tempo de cada nível e muitas outr

