

”

E-fólio B | Instruções para a realização do E-fólio

ARQUITETURA DE COMPUTADORES 21010

Considere o seguinte conjunto de dados que correspondem ao nome de estudante e à nota por este obtida num determinado trabalho:

Nome	Nota
ALICE	16
JOÃO	10
LIA	12
PAULA	15
FILIPE	18
LUCAS	12

Os dados são guardados no seguinte formato:

Parte I	Parte II
---------	----------

A Parte I corresponde a um campo de 6 words do P3 no qual é guardado o nome do estudante em ASCII de 8 bits, e a Parte II corresponde a 1 word que guarda a nota do estudante em binário.

a) **[1 valor]** Faça um programa no Assembly do P3 que execute o método da pesquisa linear e retorne a informação relativa à nota que um determinado estudante teve.

Considere os seguintes requisitos:

- a tabela de notas se encontra armazenada no endereço 5000h;
- os seis bytes a partir da posição 5050h são utilizados para colocar o nome do estudante de quem se pretende saber a nota;
- a nota correspondente ao nome indicado é retornada no registo R1.

Em relação ao nome este encontra-se justificado à esquerda ou seja a primeira letra do nome ocupa a posição mais baixa da memória, está escrito usando apenas letras maiúsculas, sem acentos. Nas posições de memória restantes, caso o nome tenha menos de 6 letras, aparece o carácter espaço (ficheiro DadosMemA.txt).

O método de pesquisa linear consiste em percorrer toda a tabela, elemento a elemento procurando o nome do estudante de quem se pretende saber a nota, e uma vez encontrado este nome, aceder e devolver a nota correspondente.

Quando um nome começa por 0, sabe-se que a lista terminou.

O programa deve retornar 0 no registo R1 caso o nome do estudante não seja encontrado.

Apresente no relatório o exemplo da procura da nota do estudante Filipe.

NOTA: O ficheiro de memória pode ser carregado utilizando a opção do Simulador Ficheiro->Carrega Memória selecionando-se depois o ficheiro DadosMemA.txt

b) **[1 valor]** O método de pesquisa linear é um método de simples implementação no entanto tem um custo de pesquisa proporcional à dimensão dos dados. Um método alternativo de pesquisa pode ser implementado recorrendo à noção de "Hash Table" (costuma traduzir-se para tabela de hash, ou seja hash não se costuma traduzir, se bem que se encontre a tradução de tabela de dispersão).

Neste caso utiliza-se uma função de hash que é aplicada a cada nome produzindo um hash value.

As funções de hash têm uma série de propriedades a que deve obedecer e são muito usadas também no âmbito da segurança e criptografia de dados, mas neste caso a propriedade que é relevante é que produz um valor de comprimento fixo ou seja o hash value é sempre do mesmo tamanho independentemente do comprimento da mensagem (neste caso o nome).

Nesta alínea pretende-se que construa um programa no Assembly do P3 que implemente uma função de hash, conhecida por Pearson Hashing (https://en.wikipedia.org/wiki/Pearson_hashing).

A função vai ter como entrada um nome da tabela de nomes (com os seis bytes para simplificar) e calcular o hash value correspondente a esse nome, que terá comprimento de 1 byte.

O hash value (h) é iniciado a 0 e é (re)calculado percorrendo os bytes da mensagem.

Para cada byte o h é atualizado para um valor dado por uma matriz de permutação T que contém aleatoriamente os valores entre 0 e 255. O índice do elemento que será atribuído a h é calculado fazendo o XOR do byte da mensagem com o valor corrente de h.

A matriz de permutações T é fornecida e fica armazenada nas posições de memória iniciadas em 4000h (ficheiro DadosMemB.txt).

c) **[1 valor]** Faça um programa no Assembly do P3 que transforme os dados tal como foram utilizados na alínea a) a partir da posição 5000h, para uma estrutura de hash table da seguinte forma:

Para cada nome deve ser calculado o hash value utilizando o código desenvolvido na alínea anterior.

Determinar o resto da divisão inteira do valor de hash por 4, obtendo assim um valor entre 0 e 3.

Os dados relativos a cada estudante deverão ser colocados iniciando nas posições de memória constantes da seguinte tabela:

Valor	Endereço
0	5500h
1	5530h
2	5560h
3	5590h

Neste caso haverá necessariamente colisões dado que vamos transformar 6 registos de dados num valor entre 0 e 3. Quando tal acontecer armazenam-se os dados do novo registo após o registo que já se encontra armazenado na zona de memória correspondente.

d) **[1 valor]** Faça um programa no Assembly do P3 que faça a procura da nota a partir de um nome de estudante, tal como aconteceu na alínea a), mas usando neste caso a hash table tal como foi sendo construída nas duas alíneas anteriores.

A pesquisa será iniciada calculando o hash value do nome, fazendo a correspondência deste para a sua zona de memória e utilizando depois a pesquisa linear apenas nos dados desta zona de memória.

Reutilize o código já implementado para a pesquisa linear na alínea a).

Apresente no relatório o exemplo da procura da nota do estudante Filipe.

NOTA: A utilização da estrutura da hash table torna a pesquisa mais eficiente mas tal é relevante para dados de grandes dimensões. Neste e-fólio trata-se de uma abordagem pedagógica exemplificativa com uma dimensão muito reduzida de dados.

BOM TRABALHO!

Regras para entrega do e-fólio B:

Forma de entrega:

Um ficheiro zip (extensão .zip e não outra como .rar) com o nome correspondente ao número de aluno.

O ficheiro zipado deverá por sua vez conter um ficheiro de Assembly do P3 por alínea, cujo nome será o número de aluno mais a letra da alínea ex: 999999a.as.

Deverá ainda ser entregue um pequeno relatório em formato pdf de até 5 páginas A4, com todos os cálculos e todas as opções tomadas na construção dos programas.

Regras relativas à escrita dos programas de Assembly:

Todos os programas devem acabar com a seguinte instrução:

```
FIM: JMP FIM          ;Fim do programa
```

Não são aceites entregas fora da plataforma Moodle.

Avaliação

Cotação:

A cotação encontra-se junto de cada uma das alíneas, entre [].

Critérios de Correção:

Funcionalidade: 50%

Simplicidade e Modularidade: 10%

Eficiência (serão contabilizados o número de instruções e ciclos de relógio): 10%

Apresentação do código (indentação e comentários): 20%

Relatório (Legibilidade e Justificação dos Resultados e das Opções): 10%

Descontos:

Trabalhos entregues que não estejam em conformidade com as regras de entrega do e-fólio B: até 10%

Código sem comentários, ou apenas com comentários a refletir o significado da instrução (exemplo MOV R1,R2 ;mover o conteúdo de R2 para R1) : até 50%

Deteção de fraude (total ou parcial): 100%

Trabalhos entregues após a data limite (o recurso de entrega estará aberto até ao início da manhã do dia seguinte, no entanto entregas após a data/hora indicada, têm desconto na nota) : 10%