Arquitectura de Computadores

Licenciatura em Engenharia Informática

2^o teste (A) – 2007/06/11 – Duração: 2h00m

Nome:_____ Número:_____

Total de páginas: 5+___ *páginas* Classificação:_____

Teste sem consulta.
A interpretação do enunciado faz parte da avaliação. Explicite nas suas respostas todas as hipóteses assumidas. Por favor, tente focalizar a sua respostas para que estas se enquadrem na zona delimitada.

Resumo da classificação:

Q-1 [1.25] = _____	Q-4 f) [0.50] = _____	Q-10 a) [1.00] = _____
Q-2 [0.75] = _____	Q-4 g) [0.50] = _____	Q-10 b) [1.00] = _____
Q-3 [1.00] = _____	Q-5 [1.00] = _____	Q-10 c) [1.50] = _____
Q-4 a) [0.25] = _____	Q-6 [2.00] = _____	Q-10 d) [1.50] = _____
Q-4 b) [0.50] = _____	Q-7 [0.50] = _____	Q-10 e) [1.00] = _____
Q-4 c) [0.75] = _____	Q-8 [1.25] = _____	Q-11 [1.00] = _____
Q-4 d) [0.75] = _____	Q-9 [1.25] = _____	**Prática [7.00]** = _____
Q-4 e) [0.75] = _____	**Teórica [13.00]** = _____	**Total [20.00]** = _____

Q-1 [1.25 val.] Explique o mecanismo de prioridades de interrupções no Intel 80x86. Na sua resposta inclua em que casos uma rotina de atendimento de interrupções pode ser ela própria interrompida.

Q-2 [0.75 val.] Que tipo de periférico pode beneficiar do uso de DMA, porquê?

Q-3 [1.00 val.] Quais são as tarefas do CPU num processo de transferência de dados por DMA?

Q-4 Admita uma arquitectura de um computador com as seguintes características: endereçamento de 28 bits, cache associativa por conjunto de 2 MBytes, onde cada conjunto tem 4 linhas com 32 Bytes, registos e bus de dados de 32 bits (4 Bytes).

a) [0.25 val.] Qual é o tamanho máximo para o espaço de endereçamento de um processo?

b) [0.50 val.] Indique para o endereço seguinte qual é o grupo, a chave do bloco de memória e o byte que é referenciado dentro desse bloco:

0110 1111 0000 0001 0000 01100 1011

grupo: _____ chave: _____ byte dentro do bloco: _____

c) [0.75 val.] Num acesso de leitura de 32 bits ao endereço da alínea anterior quantos bytes são transmitidos para o registo destino e quais são os seus endereços?

d) [0.75 val.] Um vez separados o grupo, a chave e o deslocamento, explique como encontrar na cache os bytes a transmitir e como é decidido se se trata de um *cache hit* ou *cache miss*.

e) [0.75 val.] Suponha o seguinte cenário:

- o bloco pretendido não se encontra na cache;
- a cache está cheia;
- a cache usa uma política de *write-back*;
- o acesso que se está a efectuar é de escrita.

Explique que passos são efectuados até que os bytes sejam escritos nos endereços pretendidos.

f) [0.50 val.] Considere que a cache tem um tempo de acesso de 4 ns, que memória central tem um tempo de acesso de 60 ns e que a taxa de sucesso (*hit ratio*) no acesso à cache é de 90%. Indique o tempo médio de acesso à memória.

g) [0.50 val.] Considere ainda que em média 20% do processamento é passado em acessos a memória. Calcule qual é o impacto da cache da alínea anterior na performance global do sistema.

Q-5 [1.00 val.] Numa gestão de memória virtual baseada em paginação a pedido, qual o papel desempenhado pelo bit *residente* que existe em cada entrada na tabela de páginas? Diga o que acontece quando o CPU referencia uma página cujo bit *residente* está a 0.

Q-6 [2.00 val.] Faça um esquema que ilustre os componentes e o funcionamento da MMU numa arquitectura com memória virtual. Anote o esquema com a enumeração dos passos necessários à obtenção do endereço real e use essa enumeração para elaborar a legenda do esquema.

Legenda:

Q-7 [1.25 val.] Em que é que consiste uma arquitectura super-escalar e quais são as suas vantagens e desvantagens?

Q-8 [1.25 val.] Comente a seguinte afirmação. *Em arquitecturas MIMD de memória partilhada implementadas com um único bus de sistema existe um limite a partir do qual adicionar um novo processador pode, na realidade, diminuir a performance da máquina.*

Q-9 Relembre o trabalhos práticos sobre programação de entradas/saídas. Pretende-se implementar um programa que envie o conteúdo de um ficheiro para a porta série, usando interrupções. **As perguntas devem respondidas na página seguinte.**

a) [1.00 val.] Complete a implementação das funções `ligaIntUART` que liga as interrupções na UART e `ligaIntPIC` que programa o PIC para aceitar interrupções da linha associada à porta série. Lembre-se que:

- para ligar as interrupções na UART tem de colocar o bit 3 do registo MCR a 1;
- para que as interrupções sejam desencadeadas pela recepção de um byte, o bit 0 do registo IER deve ser colocado a 1;
- para que sejam desencadeadas pela disponibilidade da porta para o envio de um novo byte, o bit 1 do registo IER deve ser colocado a 1.
- no caso do PIC, o bit correspondente à linha de IRQ deve ser colocado a 0. Por exemplo, se a linha usada pela porta série fosse a 1, deveria colocar o bit 1 a 0 no registo apropriado.

b) [1.00 val.] Complete a implementação das funções `ConfVectorInts`, que coloca a função para tratar as interrupções da porta série na posição indicada e `ReporVectorInts`, que repõe a configuração inicial do vector de interrupções.

c) [1.50 val.] Complete a implementação da função `trataInt` que trata as interrupções da porta série.

d) [1.50 val.] Complete a implementação da função `enviaSérie` que coloca o byte dado no buffer circular.

e) [1.00 val.] Complete a implementação da função `main`. Esta deve configurar o sistema para que a transmissão se possa processar, ler o conteúdo do ficheiro para um buffer circular e repor a configuração inicial do sistema.

```
#include <dos.h>
#include <stdio.h>
#include "bufcir.h"

#define PICMASK 0x21
#define PICCMD  0x20
#define EOI     0x20
#define THR     0x3F8
#define MCR     0x3FC
#define IER     0x3F9

extern unsigned char inByte( unsigned int port );
extern void outByte( unsigned int port, unsigned char byte );
void interrupt (*oldvec) (void);

void ligaIntUART() {
    unsigned char b = inByte(MCR);
    outByte(                , );
    b = inByte(IER);
    outByte(                , );
}
void desligaIntUART() { ... }

void ligaIntPIC() {
    unsigned char b;
    disable();
    b =                        ;
                               
}
void desligaIntPIC() { ... }

void interrupt trataInt() {
                               
    if(                        ) {
                               
                               
    }
    outByte(PICCMD,          );
}

void ConfVectorInts() {
                               
                               
}
```

```
void reporVectorInts() {
    [                          ]
}

void enviaSerie(unsigned char c) {
    if(bufEmpty()) {
        [                          ]
    }
    else {
        [                          ]
    }
}

int main(int argc, char *argv[]) {
    [                          ] c;
    FILE *fp;
    if (argc != 2) return -1; // Erro no numero de argumentos
    [                          ]
    fp = fopen(argv[1], "r");
    while (!feof(fp)) {
        c = fgetc(fp);
        [                          ]
    }
    fclose(fp);
    [                          ]
    [                          ]
    [                          ]
    return 0;
}
```

Q-10 [1.00 val.] Recorde agora a programação de entradas/saídas por espera activa. Complete a implementação da função `enviaSerie` que envia um byte para a porta série. Lembre-se que o bit no registo de estado (LSR) que indica que existe uma byte para ser lido é o 0, e que se pode escrever um novo byte é o 5.

```
void enviaSerie(unsigned char c) {
    unsigned char s;

    do {
       s = [                          ]
    }
    while ((s [                ]) == 0);
    outByte([                ], [                ]);
}
```