

# Fundamentos de Sistemas de Operação MIEI 2015/2016

2º teste, 7 de dezembro de 2015, 2h

Nº \_\_\_\_\_ Nome \_\_\_\_\_

*Sem consulta e sem esclarecimento de dúvidas; indique eventuais hipóteses assumidas nas sua respostas.*

**Questão 1 (2.5 valores)** Um dos problemas da gestão da memória central (RAM) em ambientes que suportam um número variável de processos carregados em memória é a *fragmentação externa*.

- a) Explique porque é que este problema ocorre quando a memória física é gerida por uma MMU com um registo base e um registo limite.
- b) E se a MMU for baseada em páginas o problema ocorre? Justifique.

**Questão 2 (2.5 valores)** Considere um sistema de gestão de memória baseado em páginas, suportando paginação a pedido. É possível executar um programa cuja imagem tenha uma dimensão superior à da memória física? Se sim, explique como.

**Questão 3 (2.5 valores)** Suponha um sistema de operação que executa numa máquina com uma MMU baseada em páginas. Considere a chamada ao sistema *fork()* usada nos sistemas operativos da família Linux / UNIX.

- a) Explique como seria preenchida a tabela de páginas do processo filho.
- b) Diga como poderia otimizar as ações descritas em a) em termos da diminuição do tempo de execução da chamada ao sistema *fork()* e também da diminuição do nº de páginas físicas ocupadas efetivamente.

**Questão 4 (2.5 valores)** Suponha um sistema de operação que executa numa máquina com uma MMU baseada em páginas em que cada página tem 4 Kbytes. Executa-se um programa que contém o seguinte fragmento:

```
#define NPAGES 20
#define PAGESIZE 4096
...
unsigned char vec[NPAGES*PAGESIZE]; // 1 char ocupa um byte na RAM
for ( i=0; i<5; i++ )
    for( j=0; j < NPAGES; j++ )
        for( k=0; k < PAGESIZE; k++ )
            vec[j*PAGESIZE+k] = (unsigned char)i;
...
```

Considere que o código acima já está carregado em RAM, que o vetor *vec* não está carregado em RAM, e que as variáveis *i*, *j* e *k* estão em registos do CPU; pode assim supor-se que as únicas faltas de página que ocorrem têm a ver com o acesso ao vetor *vec*. Diga quantas faltas de página ocorrem durante a execução do pedaço de código acima indicado quando:

- a) é possível ocupar na RAM 20 páginas com o vetor *vec*. Justifique.
- b) é apenas possível manter em RAM 5 páginas do vetor *vec*. Suponha que o algoritmo de substituição de páginas é o LRU (*least recently used*). Justifique.

**Questão 5 (3.0 valores)** Considere o sistema de ficheiros do UNIX. Explique o que se passa, em termos de alterações nos meta-dados e nas diretorias quando se

a) Muda o nome a um ficheiro com o comando

```
mv ./f1 ./f2
```

O 1º argumento é o nome antigo e o 2º argumento é o nome novo. Suponha que antes de dar o comando o ficheiro *f1* existe e o ficheiro *f2* não existe.

b) Se apaga um ficheiro com o comando

```
rm /d1/f3
```

Suponha que antes de dar o comando a diretoria *d1* e o ficheiro *f3* existem

**Questão 6 (2.0 valores)** Considere um sistema de ficheiros que usa o método indexado para atribuir blocos a um ficheiro. A localização dos blocos é definida por um vetor  $a[ ]$  com 12 endereços

- Os endereços  $a[0]$  a  $a[9]$  são endereços diretos
- O endereço  $a[10]$  é o endereço de um bloco que contém endereços de blocos (indireto simples)
- O endereço  $a[11]$  é o endereço de um bloco que contém endereços de blocos que contém por sua vez endereços de blocos (indireto duplo)

Supondo que em cada bloco do disco cabem 1024 endereços e cada bloco tem 4 Kbytes, qual é o maior tamanho de ficheiro que pode existir neste sistema de ficheiros? Justifique indicando a expressão usada no cálculo.

**Questão 7 (3 valores)** Considere o sistema de ficheiros FATX-32 com as seguintes características principais:

- o bloco 0 é o superbloco e contém o número mágico, o número de blocos NB do disco e um vetor de inteiros  $V$  sem sinal com NB posições. Cada entrada  $V[i]$  tem o seguinte conteúdo:
  - $0x00000000$  bloco  $i$  está livre
  - $0xFFFFFFFF$  bloco  $i$  é o fim da lista ligada de blocos em que o ficheiro está organizado
  - Próximo bloco do ficheiro, constituindo a lista ligada de blocos que começa na entrada da diretoria correspondente.
- O bloco1 contém a única diretoria existente. Cada entrada desta diretoria tem
  - Uma marca a dizer se está ocupada ou livre
  - O nome do ficheiro
  - Um inteiro com o comprimento do ficheiro em bytes
  - O número do bloco onde começa o ficheiro. Isto é, se este campo da entrada da diretoria é  $c$ ,  $V[c]$  é o 1º bloco do ficheiro.
- Os restantes blocos são blocos de dados

Suponha que quando um disco é montado se pretende verificar se o sistema de ficheiros está coerente. O que seria necessário fazer? Note que não se pretende corrigir as inconsistências eventualmente existentes.

**Questão 8 (2 valores)** Suponha um disco com as seguintes características:

- *seek time* médio: 12 ms
- tempo de uma rotação: 6 ms
- 500 sectores por pista; o controlador do disco tem memória suficiente para ler uma pista inteira

Apresente, justificando, uma estimativa para o tempo que demora a leitura de 20 blocos nas seguintes duas situações:

- a) os 20 blocos estão espalhados aleatoriamente pelo disco
- b) os 20 blocos estão todos contíguos na mesma pista.

Note que não precisa de apresentar um valor numérico, mas apenas as expressões que permitem calcular os dois valores.