# Fundamentos de Sistemas de Operação MIEI 2013/2014

## 2º Teste, 7 Dezembro 2013, 2 horas – versão A

Número_____ Nome _____

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

**Questão 1 (2.0 valores)** Um dos problemas da gestão da memória central em ambientes que suportam um número variável de processos carregados em memória é a *fragmentação externa*.

a) Explique em que consiste esse problema

b) Se um sistema tiver uma MMU com registo base e um registo limite está sujeito a este problema? Justifique.

c) E se a MMU for baseada em páginas? Justifique.

**Questão 2 (2.0 valores)** Um sistema em que os endereços virtuais têm 32 bits, usa uma MMU que suporta páginas com dimensão 64 KBytes (2^16). Para um dado processo em execução, as primeiras entradas da tabela de páginas são as seguintes:

| Nº página virtual | Nº página física (em base 16) |
|---|---|
| 0 | 0xCAFE |
| 1 | 0xDEAD |
| 2 | Inválida |
| 3 | 0xBEEF |

As outras entradas são todas inválidas. Indique, justificando, os endereços físicos que correspondem aos seguintes endereços virtuais. Responda "Endereço Inválido" se o endereço virtual for inválido.

a) 0x00000000

b) 0x00022001

c) 0x10001001

d) 0x0003BA11

**Questão 3 (2.5 valores)** Considere um sistema que tem uma MMU baseada em páginas e que usa paginação a pedido. O endereço virtual gerado tem os bits mais significativos a página virtual PV e nos bits menos significativos o deslocamento D. Cada entrada da tabela de páginas tem um bit V que vale 1 se a página PV está carregada na RAM e 0 se a página não está carregada em RAM.

Quando um processo gera um endereço virtual em que na entrada correspondente a PV o bit V está a 0, é gerada uma interrupção por falta de página e a MMU regista o valor PV. Supondo que a página PV pertence ao espaço de endereçamento do processo, descreva as acções do sistema operativo após a ocorrência da interrupção.

**Questão 4 (2.5 valores)** Considere o código C abaixo que faz parte de um simulador de memória virtual semelhante ao que foi desenvolvido nas aulas práticas. O sistema simulado tem PF páginas físicas e o estado de cada página física está representado da seguinte forma:

```
#define MAX_FRAMES 128
struct frame_entry{
  unsigned int referenced;    //  1 se a página foi referenciada desde que o campo
                              // foi colocado a 0; 0 se tal não aconteceu
  unsigned int virtualPage; // número da página virtual que está carregada nesta página física
}
struct frame_entry frame_table[MAX_FRAMES];
int currentPos = 0; // variável global cujo valor sobrevive entre
                    // invocações de findVictim2ndChance()
```

A função seguinte – que está incompleta - simula o algoritmo que é chamado quando não há páginas físicas livres e é preciso escolher uma página física para carregar uma página virtual. O algoritmo usado para escolher a página física que vai acolher a página é o "second chance algorithm" ou "clock". Complete o código apresentado.

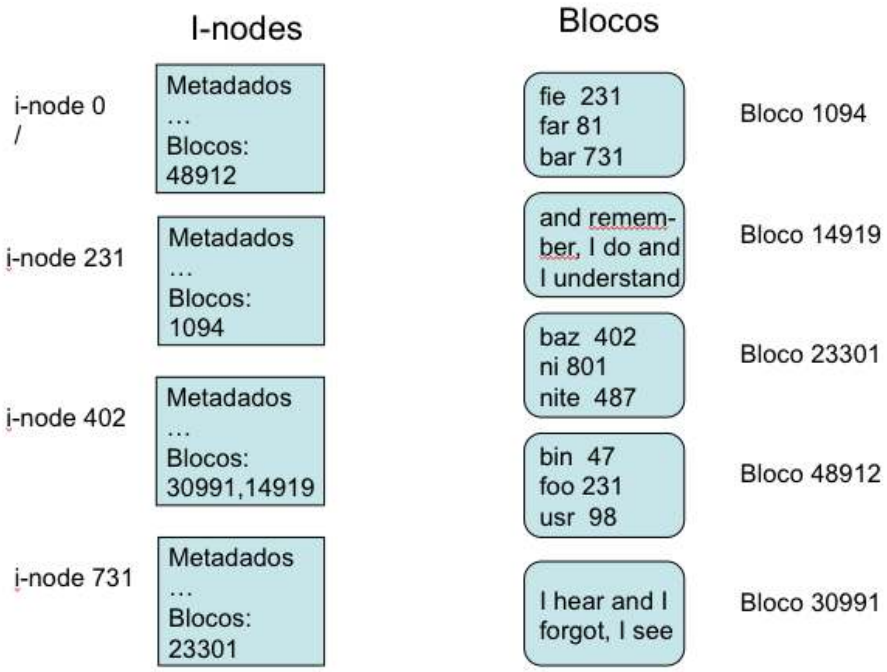
```
int findVictim2ndChance(){
    victim = -1;
    while( victim < 0){




    }
    return victim;
```

```
}
```

**Questão 5 (2.0 valores)** A figura seguinte representa uma parte de um sistema de ficheiros UNIX.




No kernel do sistema existe a indicação de que a directoria raiz corresponde ao i-node 0; a tabela de i-nodes já foi lida para RAM quando o disco foi montado. Diga que blocos de disco são lidos e por que ordem, quando um processo faz a chamada ao sistema `f = open("/foo/bar/baz", O_RDONLY)`

**Questão 6 (2.5 valores)** Para um sistema de ficheiros UNIX, indique, justificando, a sequência de leituras e escritas feitas no disco quando se executam os seguintes comandos do *shell*

a) Se muda o nome ao ficheiro /xxx. `mv /xxx /yyy`

b) Se apaga o ficheiro `/zzz`. `rm /zzz`

Neste ultimo caso, supõe-se que o *link count* do *inode* de /zzz está a 1.

**Questão 7 (1.5 valores)** Suponha uma variante do sistema de ficheiros UNIX em que os blocos têm 6Kbytes e os endereços de blocos têm 6 bytes. Cada i-node tem:

- 12 endereços directos
- 1 endereço indirecto - isto é que contém o endereço de um bloco com endereços
- 1 endereço duplamente indirecto - contém endereços de blocos que contêm endereços de blocos com endereços

Diga, justificando, qual é máxima dimensão de um ficheiro neste sistema.

**Questão 8 (1.5 valores)** Explique a diferença entre um *hard link* e um *link* simbólico. Indique o conteúdo dos respectivos i-nodes (se existirem).

**Questão 9 (1.5)** Suponha um disco com as seguintes características:

- seek time médio: 10 ms
- tempo de uma rotação: 4 ms
- 500 sectores por pista; o controlador do disco tem memória suficiente para ler uma pista inteira
- Um sector demora 10 micro-segundos a ser lido

Apresente, justificando, uma estimativa para o tempo que demora a leitura de 100 blocos nas seguintes duas situações:

a) os 100 blocos estão espalhados aleatoriamente pelo disco

b) os 100 blocos estão todos contíguos na mesma pista.

**Questão 10 (2.0 valores)** Considere o seguinte programa que usa semáforos da interface *POSIX threads*.

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>

int x = 0;
sem_t s1, s2;

void * f1(void *arg) {
   sem_wait(&s2); sem_wait(&s1); x = x*2; sem_post(&s1);
}
void * f2(void *arg) {
   sem_wait(&s1); x = x*x; sem_post(&s1);
}
void * f3(void *arg) {
   sem_wait(&s1); x = x+3; sem_post(&s2); sem_post(&s1);
}

int main(int argc, char *argv[]) {
  pthread_t p1, p2, p3;
  sem_init( &s1, 0, 1); // semaphore s1 initial value is 1
  sem_init( &s2, 0, 0); // semaphore s2 initial value is 0
  pthread_create(&p1, NULL, f1, NULL);pthread_create(&p2, NULL, f2, NULL);
  pthread_create(&p3, NULL, f3, NULL);
  pthread_join(p1, NULL); pthread_join(p2, NULL); pthread_join(p3, NULL);

  printf("%d\n", x);
  return 0;
}
```

Suponha que se executa o programa várias vezes. Diga, justificando, qual (ou quais) o(s) valor(es) finais de x.