



Universidade de Aveiro

Departamento de Eletrónica, Telecomunicações e Informática

Sistemas de Operação / Fundamentos de Sistemas Operativos

Mini-teste, modelo

(Ano Letivo de 2021-2022)

dezembro de 2021

NºMec: _____ Nome: _____ Course: _____

Na resolução deste exame, tenha em consideração o seguinte:

- **As questões são independentes entre si.** Assim, a resposta a qualquer questão deve considerar o estado do disco tal como apresentado e não aquele que resultaria da execução do código apresentado numa outra qualquer questão.
- **As questões devem ser respondidas no contexto concreto do disco apresentado.** Respostas dos tipos *se ..., então ...* ou *fica igual* não são consideradas.
- Pode responder às questões pela ordem que quiser, muito embora se responder primeiro à primeira questão ganhará uma compreensão do sistema de ficheiros que o ajudará nas respostas às restantes.
- As questões 1, 6 e as duas mais bem cotadas das restantes têm cotação de 3,5 valores cada; as restantes têm cotação de 3,0 valores cada.
- A duração do exame é de 60+15mn.
- À saída, **deve entregar** tudo o que recebeu (enunciado, folhas de resposta e rascunhos).

Considere que se criou um disco virtual sobre um ficheiro. Esse disco foi formatado como **sofs21**, usando o programa **mksofs**, e montado no diretório **/tmp/mnt/**, usando o programa **sofsmount**. Diversas operações de manipulação de ficheiros (ficheiros regulares, diretórios e atalhos) foram a seguir efetuadas sobre esse diretório (ponto de montagem).

As listagens das páginas 5 a 9 representam o estado interno de alguns blocos do disco após as operações anteriores, mostrados usando a ferramenta **showblock**. Alguns campos do superbloco e de dois *inodes* foram intencionalmente substituídos por **???**. A tabela de *inodes* é apenas parcialmente mostrada; todos os *inodes* não mostrados estão livres e limpos. Os campos **atime**, **mtime** e **ctime** não são mostrados. Para facilitar a leitura, nos campos **name** das entradas de diretório o carácter **'\0'** foi substituído por um espaço e os nomes usados não têm espaços. Os blocos não mostrados ou as partes não mostradas de blocos apenas parcialmente mostrados não são necessários para a resposta a qualquer questão.

1. Complete o preenchimento da tabela seguinte com a informação referente a todos os ficheiros (representados por um caminho absoluto) não apagados residentes no disco.

caminho absoluto (<i>path</i>)	<i>inode</i> (<i>nInode</i>)	tipo (dir/file/symlnk)	lnkcnt
/	0		

2. Nos dados apresentados sobre o estado das estruturas de dados internas do sistema de ficheiros, alguns campos foram intencionalmente substituídos por ???.

- (a) Apresente os valores dos seguintes campos do superbloco.

ntotal: ifree:
dbtotal: dbfree:

- (b) Apresente os valores dos seguintes campos dos *inodes* 3 e 5. Pode apresentar os valores usando expressões.

inode[3].blkcnt: inode[5].blkcnt:

- (c) Apresente os valores mínimos dos seguintes campos dos *inodes* 3 e 5. Pode apresentar os valores usando expressões.

inode[3].size: inode[5].size:

3. Considere que o excerto de código seguinte é executado, não tendo sido gerada nenhuma exceção.

```
uint16_t n = soAllocInode();
soHideInode(n);
```

- (a) Apresente os valores, após a execução do excerto de código, dos campos do superbloco indicados abaixo, assim como o valor da variável *n*. Se não respondeu à questão 2, considere que antes da execução *ifree* = 1000. Resposta do tipo *não foi alterado* não será considerada.

ifree: iidx: n:

- (b) Há outros campos do superbloco, além dos contemplados na alínea anterior, que sofrem alterações em consequência da execução do excerto de código. Indique quais e os valores com que ficam após a execução.

⋮
⋮
⋮

-
4. Considere que o excerto de código seguinte é executado, não tendo sido gerada nenhuma exceção.

```
int ih = soOpenInode(9);
soFreeInodeBlocks(ih, 12);
```

- (a) Apresente os valores, após a execução do excerto de código, dos campos do superbloco seguintes, sendo que `ref[*]` representa todo o array útil. Pode usar notação compactada, se aplicável. Se não respondeu à questão 2, considere que antes da execução `dbfree = 1000`. Resposta do tipo *não foi alterado* não será considerada.

```
dbfree: .....          rbm_idx: .....

retrieval_cache.idx: ..... retrieval_cache.ref[*]: .....

insertion_cache.idx: ..... insertion_cache.ref[*]: .....
```

- (b) Apresente os valores, após a execução do excerto de código, dos seguintes campos do *inode* número 9, sendo que `d[*]`, `i1[*]` e `i2[*]` representam os arrays na totalidade. Pode usar notação compactada, se aplicável. Se não respondeu à questão 2, considere que antes da execução `size = blkcnt = 1000`.

```
size: .....          blkcnt: .....

d[*]: .....          i1[*]: .....          i2[*]: .....
```

- (c) Há bloco(s) de dados alterado(s) em consequência da execução do excerto de código anterior. Indique quais e os valores com que ficam após a execução.

⋮
⋮
⋮

-
5. Considere que o excerto de código seguinte é executado, não tendo sido gerada nenhuma exceção.

```
int ih0 = soOpenInode(0);
uint16_t n1 = soGetDirenty(ih0, "bbbb");
int ih1 = soOpenInode(n1);
uint16_t n2 = soDeleteDirenty(ih1, "bbbb");
```

- (a) Que valores são armazenados nas variáveis `n1` e `n2`?

```
n1: .....          n2: .....
```

- (b) Apresente os valores, após a execução do excerto de código, dos campos do super-bloco seguintes. Se não respondeu à questão 2, considere que antes da execução `ifree = 1000` e `dbfree = 1000`.

`ifree: dbfree:`

- (c) Dos 4 blocos de dados com entradas de diretório apresentados neste exame, há um que sofre alterações em consequência da execução do excerto de código? Indique qual e os valores com que fica após a execução.

`:`
`:`
`:`
`:`
`:`

6. Considere que o excerto de código seguinte é executado, e que após a sua execução `ret` tem o valor 0.

```
#define PERM 0755
int ret = mkdir("/zzzz", PERM);
```

- (a) Que campos dos *inodes* em uso ou que ficaram em uso após a execução sofrem alterações em consequência da execução do excerto de código? Indique os novos valores dos campos que sofrem alterações. Não considere os campos `atime`, `mtime`, `ctime`, `owner` e `group`.

`:`
`:`
`:`
`:`
`:`

- (b) Que blocos de dados sofrem alterações em consequência da execução do excerto de código? Indique os novos valores dos blocos de dados que sofrem alterações.

`:`
`:`
`:`
`:`
`:`

Estado da estrutura de dados interna do disco

Disk block 0 as superblock data

Header:

Magic number: 0x50f5
Version number: 0x21
Volume name: "sofs21_disk"
Properly unmounted: no
Number of mounts: 1
Total number of blocks in the device: ???

Inodes' metadata:

Total number of inodes: 32
Number of free inodes: ???
Inode search index: 13

Inode allocation bitmap:

```
ffffe000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Inode deleted queue:

Inode deleted queue head: 0
Inode deleted queue count: 4
Queue contents:

```
1    7    11    12 (nil) (nil) (nil) (nil) (nil) (nil)
(nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
(nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
(nil) (nil) (nil) (nil) (nil) (nil)
```

Data blocks' metadata:

First block of the data block pool: 4
Total number of data blocks: ???
Number of free data blocks: ???

Reference bitmap's metadata:

First block of the bitmap table: 3
Number of blocks of the bitmap table: 1
Index of first bit position to start retrieving references: 181

Retrieval cache:

Index of the first occupied cache position: 34

Cache contents:

```
(nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
(nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
(nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
(nil) (nil) (nil) (nil) 155 156 157 158 159 160
161 162 163 164 165 166 167 168 169 170
171 172 173 174 175 176 177 178 179 180
```

Insertion cache:

Index of the first empty cache position: 59

Cache contents:

```
72 73 74 75 76 77 78 79 80 81
82 83 84 85 86 87 88 89 90 91
92 93 94 95 96 97 98 99 100 101
102 103 104 105 106 107 108 109 110 111
112 113 114 115 116 24 141 142 143 145
146 147 148 149 150 151 152 153 144 (nil)
```

Disk block 1 as inode entries

```
Inode #0
type = directory, permissions = rwxr-xr-x, lnkcnt = 4, owner = 1000, group = 1000
size in bytes = 1024, block count = 1
d[*] = 0 (nil) (nil) (nil) (nil) (nil), i1 = (nil), i2 = (nil)
-----

Inode #1
type = hidden regular file, permissions = rw-rw-r--, lnkcnt = 0, owner = 1000, group = 1000
size in bytes = 3072, block count = 2
d[*] = 139 (nil) 140 (nil) (nil) (nil) (nil), i1 = (nil), i2 = (nil)
-----

Inode #2
type = directory, permissions = rwxrwxr-x, lnkcnt = 2, owner = 1000, group = 1000
size in bytes = 1024, block count = 1
d[*] = 1 (nil) (nil) (nil) (nil) (nil), i1 = (nil), i2 = (nil)
-----

Inode #3
type = regular file, permissions = rw-rw-r--, lnkcnt = 2, owner = 1000, group = 1000
size in bytes = ???, block count = ???
d[*] = 6 7 8 (nil) (nil) (nil), i1 = (nil), i2 = (nil)
-----

Inode #4
type = directory, permissions = rwxrwxr-x, lnkcnt = 3, owner = 1000, group = 1000
size in bytes = 1024, block count = 1
d[*] = 2 (nil) (nil) (nil) (nil) (nil), i1 = (nil), i2 = (nil)
-----

Inode #5
type = regular file, permissions = rw-rw-r--, lnkcnt = 1, owner = 1000, group = 1000
size in bytes = ???, block count = ???
d[*] = 18 19 (nil) (nil) (nil) 117, i1 = 118, i2 = 126
-----

Inode #6
type = directory, permissions = rwxrwxr-x, lnkcnt = 2, owner = 1000, group = 1000
size in bytes = 1024, block count = 1
d[*] = 3 (nil) (nil) (nil) (nil) (nil), i1 = (nil), i2 = (nil)
-----

Inode #7
type = hidden directory, permissions = rwxrwxr-x, lnkcnt = 0, owner = 1000, group = 1000
size in bytes = 1024, block count = 1
d[*] = 4 (nil) (nil) (nil) (nil) (nil), i1 = (nil), i2 = (nil)
-----

Inode #8
type = regular file, permissions = rw-rw-r--, lnkcnt = 1, owner = 1000, group = 1000
size in bytes = 12, block count = 1
d[*] = 154 (nil) (nil) (nil) (nil) (nil), i1 = (nil), i2 = (nil)
-----

Inode #9
type = regular file, permissions = rw-rw-r--, lnkcnt = 1, owner = 1000, group = 1000
size in bytes = 14336, block count = 9
d[*] = (nil) (nil) 130 131 132 133, i1 = 134, i2 = (nil)
-----

Inode #10
type = symlink, permissions = rwxrwxrwx, lnkcnt = 1, owner = 1000, group = 1000
size in bytes = 12, block count = 1
d[*] = 5 (nil) (nil) (nil) (nil) (nil), i1 = (nil), i2 = (nil)
-----

Inode #11
type = hidden regular file, permissions = rw-rw-r--, lnkcnt = 0, owner = 1000, group = 1000
size in bytes = 0, block count = 0
d[*] = (nil) (nil) (nil) (nil) (nil) (nil), i1 = (nil), i2 = (nil)
-----

Inode #12
type = hidden regular file, permissions = rw-rw-r--, lnkcnt = 0, owner = 1000, group = 1000
size in bytes = 0, block count = 0
d[*] = (nil) (nil) (nil) (nil) (nil) (nil), i1 = (nil), i2 = (nil)
-----

:
```

Disk block 4 as direntries

.	0000000000
..	0000000000
	(nil)
cccc	0000000002
bbbb	0000000004
aaaa	0000000008
	(nil)
	(nil)
⋮	
	(nil)
	(nil)

Disk block 5 as direntries

.	0000000002
..	0000000000
aaaa	0000000003
dddd	0000000005
	(nil)
	(nil)
⋮	
	(nil)
	(nil)

Disk block 6 as direntries

.	0000000004
..	0000000000
cccc	0000000006
bbbb	0000000010
	(nil)
	(nil)
⋮	
	(nil)
	(nil)

Disk block 7 as direntries

.	0000000006
..	0000000004
	(nil)
aaaa	0000000009
dddd	0000000003
	(nil)
	(nil)
⋮	
	(nil)
	(nil)

Disk block 122 as references

0000:	0000000119	(nil)	(nil)	0000000120	0000000121	0000000122	(nil)	(nil)
0008:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	0000000123
0016:	0000000124	0000000125	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0024:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0032:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
:	:	:	:	:	:	:	:	:
0240:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0248:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)

Disk block 130 as references

0000:	0000000127	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0008:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0016:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0024:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0032:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
:	:	:	:	:	:	:	:	:
0240:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0248:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)

Disk block 138 as references

0000:	(nil)	(nil)	(nil)	(nil)	0000000135	0000000136	0000000137	0000000138
0008:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0016:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0024:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0032:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
:	:	:	:	:	:	:	:	:
0240:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0248:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)

Disk block 131 as references

0000:	(nil)	(nil)	(nil)	0000000128	0000000129	(nil)	(nil)	(nil)
0008:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0016:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0024:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0032:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
:	:	:	:	:	:	:	:	:
0240:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)
0248:	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)	(nil)

Disk block 3 as bitmap 32-bit words

0000:	fef3fe00	ffffffff	000000ff	00000000	00000000	ffe00000	ffffffff	ffffffff
0008:	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff
0016:	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff
0024:	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff
:	:	:	:	:	:	:	:	:
0128:	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff	ffffffff
0136:	ffffffff	00000fff	00000000	00000000	00000000	00000000	00000000	00000000
0144:	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0152:	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
:	:	:	:	:	:	:	:	:
0248:	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

Assinatura das funções referenciadas neste exame

```
int soOpenInode(uint16_t in);
uint16_t soAllocInode();
void soHideInode(uint16_t in);
void soFreeInodeBlocks(int ih, uint32_t fibn);
uint16_t soGetDirentry(int pih, const char *name);
uint16_t soDeleteDirentry(int pih, const char *name);
int soMkdir(const char *path, mode_t mode);
```

Declaração das estruturas de dados internas do sofs21

```
#define BlockSize 1024U                                /** block size (in bytes) */
#define IPB (BlockSize / sizeof(SOInode))              /** number of inodes per block (16) */
#define DPB (BlockSize / sizeof(SODirEntry))          /** number of direntries per block (16) */
#define RPB (BlockSize / sizeof(uint32_t))            /** number of references per block (256) */
#define NullBlockReference 0xFFFFFFFF                /** null reference to a data block */
#define NullInodeReference 0xFFFF                    /** null reference to an inode */

#define PARTITION_NAME_LEN 19                          /** maximum length of volume name */
#define REF_CACHE_SIZE 60                             /** size of caches in superblock for inode references */
#define MAX_INODES (100*32)                          /** size of caches in superblock for inode references */

struct SOSuperBlock                                    /** Definition of the superblock data type. */
{
    uint16_t magic;                                    /** magic number - file system identification number */
    uint8_t version;                                  /** version number */
    uint8_t mntstat;                                   /** mount status (1: properly unmounted; 0: otherwise) */

    char name[PARTITION_NAME_LEN + 1]; /** volume name */

    uint32_t ntotal;                                   /** total number of blocks in the device */

    uint16_t itotal;                                   /** total number of inodes */
    uint16_t ifree;                                    /** number of free inodes */
    uint16_t iidx;                                     /** number of last allocated inode */
    uint32_t ibitmap[MAX_INODES/32] /** bitmap representing inode allocation states */

    uint16_t iqueue[DELETED_QUEUE_SIZE]; /** (circular) queue of deleted/hidden inodes */
    uint8_t iqhead;                                    /** index of oldest reference in iqueue */
    uint8_t iqcount;                                   /** number of elements in iqueue */

    uint32_t dbp_start;                                /** physical number of the block where the data zone starts */
    uint32_t dbotal;                                   /** total number of data blocks */
    uint32_t dbfree;                                   /** number of free blocks in data zone */

    uint32_t rbm_start; /** physical number of the block where the reference bitmap starts */
    uint16_t rbm_size;  /** number of blocks the reference bitmap comprises */
    uint32_t rbm_idx;   /** bit number where to start retrieving references from bitmap */

    struct ReferenceCache /** cache of references to free data blocks */
    {
        uint32_t idx;                /** index of first free/occupied cell */
        uint32_t ref[REF_CACHE_SIZE]; /** the cache itself */
    };

    ReferenceCache retrieval_cache; /** retrieval cache of references to free data blocks */
    ReferenceCache insertion_cache; /** insertion cache of references to free data blocks */
};
```

```

#define DIRECTORY_SLOT 30                /** maximum length of a file name (in characters) */

struct SODirectorySlot                  /** Definition of the directory slot data type. */
{
    uint16_t in;                        /** the associated inode number */
    char nameBuffer[DIRECTORY_SLOT];    /** buffer to hold name or part of it of a directory entry */
};

```

```

#define N_DIRECT 6                      /** number of direct block references in the inode */

struct SOInode                         /** Definition of the inode data type. */
{
    uint16_t mode;                      /** inode mode: it stores the file type and permissions. */
    uint16_t lnkcnt;                   /** link count: number of directory entries pointing to the inode */
    uint32_t owner;                    /** user ID of the file owner */
    uint32_t group;                    /** group ID of the file owner */
    uint32_t size;                     /** file size in bytes: */
    uint32_t blkcnt;                   /** block count: total number of blocks used by the file */

    uint32_t atime;                    /** time of last access to file information */
    uint32_t mtime;                    /** time of last change to file information */
    uint32_t ctime;                    /** time of last change to inode information */

    uint32_t d[N_DIRECT];              /** direct references to the first data blocks with file's data */
    uint32_t i1;                       /** reference to a block that extend the \c d array */
    uint32_t i2;                       /** reference to a block that extends the \c i1 array */
};

```
