Mini Teste

Fundamentos de Sistemas Operativos / Sistemas de Operação

Ano Lectivo de 2019/2020

27/10/2019

Nº Mec: _____ **Nome:** _____

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 do tipo *se ..., então ...* 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 `sofs19`, 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 directório o carácter `'\0'` foi substituído por um espaço. Há blocos apenas parcialmente mostrados. A parte omissa não é necessária para a resposta a qualquer questão. O mesmo acontece com os blocos não mostrados.

---

1. Complete o preenchimento da tabela seguinte com a informação referente a todos os ficheiros não apagados residentes no disco.

caminho absoluto (***path***)	**nº do *inode*** (***nInode***)	**tipo** (**dir/file/symlink**)	**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. Sabe-se que há 17 blocos de dados com referências de blocos de dados livres.

   `ntotal`: ..... `it_size`: .....

   `dz_total`: ..... `dz_free`: .....

   (b) Apresente os valores dos seguintes campos dos *inodes* 6 e 3.

   `inode[6].blkcnt`: ..... `inode[3].blkcnt`: .....

   (c) Apresente os valores mínimos dos seguintes campos dos *inodes* 6 e 3.

   `inode[6].size`: ..... `inode[3].size`: .....

---

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

```
uint32_t n = soAllocInode(S_IFREG);
soFreeInode(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`.

   `itotal`: ..... `ifree`: ..... `n`: .....

   (b) Há outros campos do superbloco, além dos contemplados na alínea anterior, que sofrem alterações em consequência da execução do excerto de código. Qual(is) e que alteração(ões) sofre(m)? ⋮

   ⋮

   ⋮

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

```
int ih = soOpenInode(3);
soFreeFileBlocks(ih, 200);
```

   (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 `dz_free = 1000`.

   `dz_free`: ..... `head_idx`: ..... `tail_idx`: .....

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

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

   (b) Apresente os valores, após a execução do excerto de código, dos seguintes campos do *inode* número 3, 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[*]`: .....

   (c) Há um bloco da zona de dados alterado em consequência da execução do excerto de código anterior. Qual? Que alterações sofre?

   ⋮
   ⋮
   ⋮

---

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

```
int ih0 = soOpenInode(0);
uint32_t n1 = soGetDirEntry(ih0, "aaaa");
int ih1 = soOpenInode(n1);
uint32_t n2 = soDeleteDirEntry(ih1, "gggg");
```

   (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 superbloco seguintes. Se não respondeu à questão 2, considere que antes da execução `dz_free = 1000`.

   `ifree`: ..... `dz_free`: .....

   (c) Dos 3 blocos de dados com entradas de diretório apresentados neste exame, qual ou quais sofrem alterações em consequência da execução do excerto de código? Que alterações sofrem?

   ⋮
   ⋮
   ⋮
   ⋮

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 = soMkdir("/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? Que alterações sofrem? Não considere os campos `atime`, `mtime` e `ctime`.

⋮
⋮
⋮
⋮

(b) Que blocos da zona de dados sofrem alterações em consequência da execução do excerto de código? Que alterações sofrem?

⋮
⋮
⋮
⋮

Estado da estrutura de dados interna do disco

Disk block 0 as superblock data

```
Header:
   Magic number: 0x50f5
   Version number: 0x2019
   Volume name: "mini-teste-modelo"
   Properly unmounted: no
   Number of mounts: 2
   Total number of blocks in the device: ???
Inodes' metadata:
   Number of blocks of the inode table: ???
   Total number of inodes: 24
   Number of free inodes: 16
   Head of list of free inodes: 10
   Tail of list of free inodes: 1
Data blocks' metadata:
   First block of the data zone: 4
   Total number of data blocks: ???
   Number of free data blocks: ???
   Head reference data block: 1
   First occupied position on head reference data block: 1
   Tail reference data block: 17
   First empty position on tail reference data block: 235
   Head cache:
      Index of the first occupied cache position: 61
      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) (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) 79 80 81
   Tail cache:
      Index of the first empty cache position: 38
      Cache contents:
         30 44 45 46 47 48 49 50 51 52
         53 54 55 56 57 58 59 60 61 62
         63 64 65 66 67 68 69 70 71 72
         73 74 75 76 77 43 18 19 (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) (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) (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)
         (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)
```

Disk block 1 as inode entries

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

```
----------------
Inode #1
type = free, permissions = ---------, lnkcnt = 0, owner = 0, group = 0
size in bytes = 0, block count = 0, next = (nil)
 d[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i1[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i2[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
----------------
Inode #2
type = directory, permissions = rwxr-xr-x, lnkcnt = 2, owner = 1000, group = 1000
size in bytes = 1024, block count = 1
 d[*] = 20 (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i1[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i2[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
----------------
Inode #3
type = regular file, permissions = rw-r--r--, lnkcnt = 1, owner = 1000, group = 1000
size in bytes = ???, block count = ???
 d[*] = (nil) (nil) 28 29 (nil) (nil) (nil) (nil)
i1[*] = 31 39 (nil) (nil) (nil) (nil) (nil) (nil)
i2[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
----------------
Inode #4
type = regular file, permissions = rw-r--r--, lnkcnt = 1, owner = 1000, group = 1000
size in bytes = 4096, block count = 4
 d[*] = 22 23 24 78 (nil) (nil) (nil) (nil)
i1[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i2[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
----------------
Inode #5
type = directory, permissions = rwxr-xr-x, lnkcnt = 2, owner = 1000, group = 1000
size in bytes = 1024, block count = 1
 d[*] = 21 (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i1[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i2[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
----------------
Inode #6
type = regular file, permissions = rw-r--r--, lnkcnt = 2, owner = 1000, group = 1000
size in bytes = ???, block count = ???
 d[*] = 25 26 (nil) (nil) (nil) (nil) (nil) (nil)
i1[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i2[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
----------------
Inode #7
type = regular file, permissions = rw-r--r--, lnkcnt = 1, owner = 1000, group = 1000
size in bytes = 12, block count = 1
 d[*] = 35 (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i1[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i2[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
```

Disk block 2 as inode entries

```
Inode #0
type = symlink, permissions = rwxrwxrwx, lnkcnt = 1, owner = 1000, group = 1000
size in bytes = 12, block count = 1
 d[*] = 27 (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i1[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i2[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
----------------
Inode #1
type = free, permissions = ---------, lnkcnt = 0, owner = 0, group = 0
size in bytes = 0, block count = 0, next = 1
 d[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i1[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i2[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
----------------
```

```
Inode #2
type = free, permissions = ---------, lnkcnt = 0, owner = 0, group = 0
size in bytes = 0, block count = 0, next = 11
 d[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i1[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i2[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
----------------
          ⋮
Inode #7
type = free, permissions = ---------, lnkcnt = 0, owner = 0, group = 0
size in bytes = 0, block count = 0, next = 16
 d[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i1[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i2[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
```

Disk block 3 as inode entries

```
Inode #0
type = free, permissions = ---------, lnkcnt = 0, owner = 0, group = 0
size in bytes = 0, block count = 0, next = 17
 d[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i1[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i2[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
----------------
          ⋮
Inode #7
type = free, permissions = ---------, lnkcnt = 0, owner = 0, group = 0
size in bytes = 0, block count = 0, next = 9
 d[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i1[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
i2[*] = (nil) (nil) (nil) (nil) (nil) (nil) (nil) (nil)
```

Disk block 4 as direntries

```
.                           0000000000
..                          0000000000
                            (nil)
gggg                        0000000002
aaaa                        0000000005
bbbb                        0000000006
                            (nil)
         ⋮
                            (nil)
```

Disk block 24 as direntries

```
.                           0000000002
..                          0000000000
aaaa                        0000000003
                            (nil)
bbbb                        0000000008
cccc                        0000000007
                            (nil)
         ⋮
                            (nil)
```

Disk block 25 as direntries

```
.                           0000000005
..                          0000000000
                            (nil)
                            (nil)
ffff                        0000000004
gggg                        0000000006
                            (nil)
         ⋮
                            (nil)
```

Disk block 5 as references

```
0000: 0000000002 0000000082 0000000083 0000000084 0000000085 0000000086 0000000087 0000000088
0008: 0000000089 0000000090 0000000091 0000000092 0000000093 0000000094 0000000095 0000000096
0016: 0000000097 0000000098 0000000099 0000000100 0000000101 0000000102 0000000103 0000000104
        ⋮
0240: 0000000321 0000000322 0000000323 0000000324 0000000325 0000000326 0000000327 0000000328
0248: 0000000329 0000000330 0000000331 0000000332 0000000333 0000000334 0000000335 0000000336
```

Disk block 20 as references

```
0000: 0000000017 0000003907 0000003908 0000003909 0000003910 0000003911 0000003912 0000003913
0008: 0000003914 0000003915 0000003916 0000003917 0000003918 0000003919 0000003920 0000003921
0016: 0000003922 0000003923 0000003924 0000003925 0000003926 0000003927 0000003928 0000003929
        ⋮
0240: 0000004146 0000004147 0000004148 0000004149 0000004150 0000004151 0000004152 0000004153
0248: 0000004154 0000004155 0000004156 0000004157 0000004158 0000004159 0000004160 0000004161
```

Disk block 21 as references

```
0000:   (nil)    0000004162 0000004163 0000004164 0000004165 0000004166 0000004167 0000004168
0008: 0000004169 0000004170 0000004171 0000004172 0000004173 0000004174 0000004175 0000004176
0016: 0000004177 0000004178 0000004179 0000004180 0000004181 0000004182 0000004183 0000004184
        ⋮
0232: 0000004393 0000004394 0000004395   (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 35 as references

```
0000: 0000000032 0000000033 0000000034   (nil)      (nil)      (nil)      (nil)      (nil)
0008:   (nil)    0000000036 0000000037 0000000038   (nil)      (nil)      (nil)      (nil)
0016:   (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 43 as references

```
0000: 0000000040 0000000041 0000000042   (nil)      (nil)      (nil)      (nil)      (nil)
0008:   (nil)      (nil)      (nil)      (nil)      (nil)      (nil)      (nil)      (nil)
0016:   (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 31 as ASCII

```
0000: .  .  /  a  a  a  a  /  f  f  f  f  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0064: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
        ⋮
0960: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0992: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Disk block 39 as ASCII

```
0000: .  .  /  g  g  g  g  /  f  f  f  f  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0064: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
        ⋮
0960: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0992: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Assinatura das funções referenciadas neste exame

```
uint32_t soAllocInode(uint32_t type);
void soFreeInode(uint32_t in);
int soOpenInode(uint32_t in);
void soFreeFileBlocks(int ih, uint32_t ffcn);
uint32_t soGetDirEntry(int pih, const char *name);
uint32 soDeleteDirEntry(int pih, const char *name);
int soMkdir(const char *path, mode_t mode);
```

Declaração das estruturas de dados internas do sofs19

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

```
#define PARTITION_NAME_SIZE 21    /** maximum length of volume name */
#define HEAD_CACHE_SIZE 64        /** size of caches in superblock for inode references */
#define TAIL_CACHE_SIZE 170       /** size of caches in superblock for block references */

struct SOSuperBlock               /** Definition of the superblock data type.  */
{
    uint16_t magic;               /** magic number - file system identification number */
    uint16_t version;             /** version number */

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

    uint8_t mntstat;              /** mount status (1: properly unmounted; 0: otherwise) */
    uint8_t mntcnt;               /** number of mounts since last file system check */

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

    uint32_t it_size;             /** number of blocks that the inode table comprises */
    uint32_t itotal;              /** total number of inodes */
    uint32_t ifree;               /** number of free inodes */
    uint32_t ihead;               /** number of fist free inode */
    uint32_t itail;               /** number of last free inode */

    uint32_t dz_start;            /** physical number of the block where the data zone starts */
    uint32_t dz_total;            /** total number of data blocks */
    uint32_t dz_free;             /** number of free blocks in data zone */
    uint32_t head_blk;            /** number of head reference data block */
    uint32_t head_idx;            /** first occupied position in head reference data block */
    uint32_t tail_blk;            /** number of tail reference data block */
    uint32_t tail_idx;            /** first empty position in tail reference data block */

    struct HeadCache              /** head cache of references to free data blocks */
    {
        uint32_t idx;
        uint32_t ref[HEAD_CACHE_SIZE];
    } head_cache;

    struct TailCache              /** tail cache of references to free data blocks */
    {
        uint32_t idx;
        uint32_t ref[TAIL_CACHE_SIZE];
    } tail_cache;
};
```

```
#define SOFS19_MAX_NAME 27                  /** maximum length of a file name (in characters) */

struct SODirEntry                           /** Definition of the directory entry data type. */
{
    uint32_t in;                            /** the associated inode number */
    char name[SOFS19_MAX_NAME + 1];         /** the name of a file (NULL-terminated string) */
};
```

```
#define INODE_FREE 0001000          /** flag signaling inode is free */
#define N_DIRECT 8                  /** number of direct block references in the inode */
#define N_INDIRECT 8                /** number of indirect block references in the inode */
#define N_DOUBLE_INDIRECT 8         /** number of double indirect 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 */

    union                           /** time of last access to file information / next free inode */
    {
        uint32_t atime;              /** time of last access to file information */
        uint32_t next;               /** next free inode */
    };
    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[N_INDIRECT];         /** references to blocks that extend the \c d array */
    uint32_t i2[N_DOUBLE_INDIRECT];  /** references to a block that extends the \c i1 array */
};
```
