1) A UNIX system-V file system has a block size of 2 Kbytes, inodes with 12 block direct addresses, one single indirect address, one double indirect address and one triple indirect address. Moreover, it uses 4 bytes for representing those addresses. How many blocks are necessary to store a file of size 2.5Mbytes?

A file of 2.5 MB occupies in that file system 2.5*1024Kytes/2Kbytes=1280 blocks. Each index block can store 2048/4 bytes= 512 indexes (block numbers). Therefore, it is necessary:

- **The 12 data blocks of the direct addresses.**
- **An indirect index block, "single indirect" block (specified with the single indirect address of the inode).**
- **512 data blocks whose addresses are stored in the previous indirect index block.**
- **An index block ("double indirect" block, first level of indirection).**
- **2 blocks for index blocks in the indexing second level (whose block numbers are stored in the previous block).**
- **512+244 data blocks, whose addresses are stored in the previous index blocks (second level index blocks).**

2) What is the number of necessary disk accesses, at minimum, in a UNIX system-V file system, for performing the sentence ***fd = open ("so/practicas/p1/practica1.c", RD_ONLY);*** It is supposed that the Buffer Cache is empty and that the inode of directory "." is already in memory.

If it is supposed that the necessary directory entries are always located in the first block, at least 5 accesses will be performed:

1. **Read the first block of the working directory for knowing the inode of *./so***
2. **Read the inode of *./so***
3. **Read the first block of the directory *./so* in order to know the inode of *practicas***
4. **Read the inode of *./so/practicas***
5. **Read the first block of the directory *./so/practicas* to known the inode of *p1***
6. **Read the inode of *./so/practicas/p1***
7. **Read the first block of the directory *./so/practicas/p1* to known the inode of *practica1.c***
8. **Read the inode of *./so/practicas/p1/practica1.c***

Moreover, if it is also supposed that the different inodes are always located in the same block of the Inode List (possible in a System-V file system), only 1 access would be necessary in the Inode List (plus the 4 accesses to the Data Area)

3) A UNIX file system has an inode size of 64 bytes, a block size of 2 Kbytes and its Inode List occupies 2048 blocks. How many blocks does a bit map require for representing the free inodes and assigned inodes?

Number of inodes: **2048 bytes/64 bytes=32 (inodes per block)**
2048*32=2^{16} inodes→ 2^{16}bits are necessary→ 4 blocks are necessary for the bit map (each block has 16Kbits).

4) In the following code, you have to include the code of a function “redirection_of_errors”, so that the error messages produced with the library function *perror* (that sends the information to the standard error output device) are stored in the disk, in the current working directory, in a file with the name “error_register”

```
#include <stdio.h>
#include <fcntl.h>
main (int argc, char *argv[]) {
int fd;
redirection_of_errors ();
fd = open(argv[1], O_RDONLY);
if (fd == -1) {
        perror("\nerror in open");
        exit(1);
       }
}
```

In all the code lines it is necessary to specify whether there is a call to a library function or a system call to the OS.

```
redirection_of_errors (){
int fd = open(“error_register ", O_WRONLY | O_APPEND); // open is a system call
close(2); // close is a system call
dup(fd); // dup is a system call
}
```