SURNAME		NAME		Group
ID		Signature		

- **Keep the exam sheets stapled.**
- **Write your answer inside the reserved space.**
- **Use clear and understandable writing. Answer briefly and precisely.**
- **The exam has 8 questions, everyone has its score specified.**

1. Let's consider a system that manages 1200 KBytes of main memory relying on contiguous allocation with variable partitions. The hole allocation policy is Worst Fit with compaction. The compaction algorithm minimizes shifts in memory until finding enough space to allocate the process that requests memory at every moment, always choosing to move processes towards the highest addresses. Consider the initial allocation state shown in the following table:

0 1200KB-1

SO 80KB	P1 160KB	HOLE 400KB	P2 120KB	HOLE 440KB

(1,0 points = 0,6 + 0,4)

a) Manage main memory, applying compaction if necessary, and indicate the base address assigned to each process when the indicated sequence of process arrivals happens, from left to right:

	Initial state	P3 arrives Requests 280K	P4 arrives Requests 200K	P5 arrives Requests 280K	P6 arrives Requests 80K
Base address of P1					
Base address of P2					
Base address of P3					
Base address of P4					
Base address of P5					
Base address of P6					

b) Explain what type of fragmentation appears in variable partitions. If it happens at some point in section a) then estimate its amount.

2. In relation to page sharing in a paging system, indicate for every one of the following statements if it is true (T) or false (F) :

Note. A wrong answer voids a correct answer. **(0,8 points)**

2	STATEMENT	T / F
	There are page descriptors belonging to different processes that contain the same frame id corresponding to shared pages	
	The copy-on-write technique allows to share pages with writing permission	
	Processes that share pages allocated in memory, also share the page table	
	A page loaded in memory in a region with permissions r-xp can be shared by several processes	
	The probability to reach thrashing increases when increasing the number of shared pages between processes	

3. A system has 24-bit both physical and logical memory addresses. Page size is 4-KByte and it manages virtual memory through paging. Free frames are assigned in increasing order, the replacement algorithm applied is LRU with LOCAL scope .

(1,2 points = 0,3 + 0,9)

3

a) Obtain the reference string corresponding to the following sequence of logical addresses (in hexadecimal) emitted by processes A and B during their execution: A:40000, A:40014, B:80000, B:80024, B:60030, A:60034, B:80280, B:4060C, A:20000, A:40C10, B:60F24

b) The system assigns frames 0 and 1 to process A and frames 2, 3, 4 to process B, all of them are initially are empty. Complete the following table with the content evolution of main memory for the reference string obtained in the previous section. Indicate the number of page faults produced.

Frame									
0 (A)									
1 (A)									
2 (B)									
3 (B)									
4 (B)									

Number of page faults:

4. A system with demand paging has two levels paging. The first level table has16 page descriptors, logical and physical addresses are both 24-bit wide, and page size is 4-Kbyte. This system assigns 3 frames to each process and it is executing process A. At time t = 10, process A has frames 0xf00, 0xf01 and 0xf02 occupied by pages 0x001, 0xf2f and 0x11a respectively .

(1,6 points = 0,4 + 0,4 + 0,8)

4

a) Describe the structure of the logical addresses and physical addresses in this system, as well as the size in bits of each one of their fields.

b) Explain the content of page descriptors for process A that have the valid bit set to 1 at time t=10.

c) Complete the following table with the corresponding logical or physical addresses, which could have been issued or accessed for process A at time t = 10, explaining if the given values are possible.

Logical address	Physical address	Is it possible? Explain why
	0xf01f02	
	0xf02f01	
0x11a13b		
0x13b11a		

5. Complete the C program in section a) with the necessary POSIX primitives (one on each line with underlined number) for a parent process to read the contents of the "message.txt" file and send 2 concatenated copies of it to its child process through a pipe. The child process will forward the content that it receives from the pipe to the standard output relying on cat program.
Note. cat, without arguments, writes everything it reads from the standard input into the standard output).

(1,4 points= 0,8 + 0,6)

5 **a)**

```
1   #include <all_needed>
2   #define SIZE 80
3   #define MODE O_RDONLY
4   int main( int argc, char **argv ){
5       int fd, fdp[2], readbytes, ncopy;
6       char buffer[SIZE];
7
8       if (fork() == 0) { // child
9
10          close(fdp[0]);
11
12          execlp("cat","cat", NULL);
13      }
14      else { // parent
15          close(fdp[0]);
16
17          for(ncopy=0; ncopy<2; ncopy++) {
18              while((readbytes= read(fd, buffer, SIZE)) > 0){
19                  write(                          ); /*complete*/
20              }
21              lseek(          ); /*complete*/
22          }
23
24
25      }
26      wait(NULL);
27      exit(0);
28  }
```

b) Fill the child's file descriptor table, at the time when line 12 is executed. Do the same for the parent process in line 17. The tables have to be compliant with the requirements and the implementation of section a).

Child process	
0	
1	
2	
3	
4	
5	

Parent process	
0	
1	
2	
3	
4	
5	

6. Given the following listing of a directory in a POSIX system :

```
i-node  permissions links    user   group   size     date      name
37093377 drwxr-xr-x  3       marta   disca   4096 dic 11 11:57 .
32448485 drwxr-xr-x  3       marta   disca   4096 dic 11 12:02 ..
32448767 -rwsr-xr-x  1       marta   disca 141528 dic 11 10:31 cp2
33373385 dr-xrwxr-x  2       marta   disca   4096 dic 11 12:02 dir1
32448804 lrwxrwxrwx  1       marta   disca      4 dic 11 10:35 dir2 -> dir1
32448793 -r--r--r--  1       marta   disca    337 dic 11 10:33 f1
32448802 -rw-r--r--  3       marta   disca    402 dic 11 10:33 f2
32448802 -rw-r--r--  3       marta   disca    402 dic 11 10:33 f3
32448803 lrwxrwxrwx  1       marta   disca      2 dic 11 10:34 f4 -> f3
```

(1,4 points = 0,8 + 0,6)

6

a) Program cp2 is similar to Linux command cp. When the last parameter of cp2 is a directory it copies the indicated files in this directory (creating the files if they do not exist). Assume that directory dir1 is empty, containing only "." and "..". In case of success indicate what are the permissions that are checked and, in case of error, what is the permission that fails and why.

(UID,GID)	COMMAND	DOES IT WORK?	EXPLANATION
(pepe, disca)	cp2 f1 dir1		
(juan, fso)	cp2 f1 dir2		
(ana, alum)	cp2 f1 f2		

b) Explain the value on the links column for every entry, indicating what is or may be the reason for that entry to have the given number of links.

NAME	LINKS	EXPLANATION
.	3	
dir1	2	
f2	3	

7. A file system is organized in blocks of 512 bytes, with block pointers of 16 bits. In this system there is a 32 KByte file named "example.txt. For every of the indicated allocation methods, obtain the number of accesses to data blocks needed to read Byte 16900 in the file considered. Explain your answer.

(1,0 points = 0,5 + 0,5)

7	**a)** Linked allocation
	b) Indexed allocation

8. A 64-MByte disk is formatted with a Minix file system, with the following specs:

- Boot block and superblock: 1 block each one
- 32-Byte i-nodes with 7 direct pointers, 1 indirect,1 double indirect
- 16-bit pointers to zone
- 16-byte directory entries: 2 Bytes for i-node, 14 Bytes for name
- 1 zone = 1 block = 1 KByte

(1,6 points = 0,3 + 0,8 + 0,5)

8	**a)** Obtain the maximum number of i-nodes that this system can have. Explain your answer.

b) Suppose the disk is formatted by reserving space in the header for a total of 32768 i-nodes (32 K i-nodes). Calculate the number of blocks occupied by each element of the file system, indicated below.

Boot	Super block	i-nodes bit map	Zones bit map	i-nodes	Data zones

c) Suppose that, at any given moment, there are a total of 32768 nodes-i (32K nodes-i) occupied on the disk with a single directory, the root directory, and regular files all of 1KByte in size.Obtain the number of data zones that will be occupied.