

<b>Operating Systems</b>	<b>Final Exam</b>	
<b>Name:</b>		

1. (0.8 pts) Answer **true** or **false** (wrong answers count negatively):

- a) The Process Control Block (**PCB**) serves to decide the order of execution among running processes.
- b) **Paging** memory virtualization does not require any memory capacity overhead to work.
- c) In a system that uses **paging**, there can be external fragmentation in memory, but no internal fragmentation.
- d) **Thrashing** consists of considerably increasing the number of virtual pages devoted to a process to speed up the translation process.
- e) Paging translates virtual addresses slower than **segmentation**.
- f) The main advantage of using a **TLB** is to speed up the address translation process in paging virtualization.
- g) Using a multi-level page table increases the latency of a hit in the TLB.
- h) The **trap table** can only be modified in **kernel** mode, and never in user mode.

2. (1 pts) What is a **context switch**? Describe what the Operating System and the Hardware should do to perform a context switch correctly.

3. (0.7 pts) **Timer interrupts** are a core mechanism of Operating Systems. Why?

4. (1 pts) Describe the process (hard drive accesses) that a simple file system must follow, like the one seen in theory, to **open** an existing file `"/proc/cpuinfo"`.

5. (0.8 pts) Describe concisely what priority inversion means in the execution of several processes. Give an example of how this problem can arise and how to solve it.

6. (1 pts) We are given multiple black boxes that are supposed to be RAID controllers. Each RAID controller is connected to an array of four identical hard drives (disk0-disk3). Whenever we launch a request to the RAID controller, we capture how it translates to different requests to the hard drives (sometimes in parallel). For each question, an original request to the RAID logical disk is given, as well as how it translates to the physical drives. Which RAID level matches with each one (RAID 0, RAID 1, RAID 4 or RAID 5). Explain your answer in detail. **Note:** If there are multiple RAID levels that match one translation, give all possible answers.

- a) write(0) → RAID [write(disk0, blk0), write(disk1,blk0)] in parallel  
write(1) → RAID [write(disk2, blk0), write(disk3,blk0)] in parallel  
write(2) → RAID [write(disk0, blk1), write(disk1,blk1)] in parallel  
write(3) → RAID [write(disk2, blk1), write(disk3,blk1)] in parallel
  
- b) read(0) → RAID [read(disk0,blk0)]
  
- c) write(0) → RAID [read(disk0, blk0), read(disk3, blk0) ] in parallel, followed by  
RAID [write(disk0, blk0), write(disk3, blk0) ] in parallel  
write(3) → RAID [read(disk0, blk1), read(disk3, blk1) ] in parallel, followed by  
RAID [write(disk0, blk1), write(disk3, blk1) ] in parallel

7. (1 pts) There is a 500GB hard drive in which we have installed a very simple Unix file system like the one seen in theory. In this file system, each inode has 10 direct pointers, a single pointer for indirect blocks and a single pointer to doubly-indirect blocks. Assume that each pointer is 32 bits in size and the file system supports 8KB blocks.

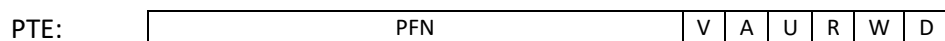
- a) What is the maximum file size supported in this file system? Explain your answer in detail.
- b) Why don't we use only doubly-indirect pointers? Pros and Cons.
- c) Name at least two ways we can change this simple file system to support larger files. What are the consequences of those changes?

8. (1.2 pts) Consider an Operating System with MLFQ (Multi-Level Feedback Queue) scheduler with three priority queues (Q1, Q2 and Q3). The time slice of each queue is: 4ms in Q1, 8ms in Q2 and 12ms in Q3. Every 60ms there is a priority boost in the MLFQ scheduler. Any job executing in the system uses the same I/O device, which can only attend one process at a time. Assume that the context switches complete in no time (0ms) for the purpose of this exercise. Get the execution scheduling of three jobs (A, B and C) running simultaneously until completion and obtain the turnaround time and the response time of each job. The jobs have the following behaviour:

<p>A:</p> <ul style="list-style-type: none"> <li>- Arrives at t=0ms.</li> <li>- Requires 20ms of CPU.</li> <li>- I/O every 4ms of CPU. I/O operations last for 12ms.</li> </ul>	<p>B:</p> <ul style="list-style-type: none"> <li>- Arrives at t=2ms.</li> <li>- Require 40ms of CPU.</li> <li>- I/O every 12ms of CPU. I/O operations last for 12ms.</li> </ul>	<p>C:</p> <ul style="list-style-type: none"> <li>- Arrives at t=6ms.</li> <li>- Require 60ms of CPU.</li> <li>- No I/O.</li> </ul>
---	---	--



9. (1 pts) Consider a system that only uses paging as virtualization of memory, and which only uses one-level page tables (lineal). The address space devoted to each process is 64GB (36 bits) and the page size is 1KB. Each page table entry (PTE) has the following information: page frame number (PFN), a valid bit (V), an access bit (A), a user/kernel bit (U), two permission bits (RW) and a dirty bit (D). If the system allows up to 64GB of physical memory.



- a) How many entries does a page table have in this system? Is this always the case?
- b) How many pages does a page table occupy in memory?
- c) How much memory do the page tables occupy if there are 50 processes running simultaneously in the system.

10. (1.5 pts) Consider a system with paging memory virtualization (without segmentation) and multi-level page tables (two levels). Each virtual address is 15 bits long, the page size is 32 bytes and the system has 4KB physical memory (12 bits physical address). Each page directory entry (PDE) or page table entry (PTE) has the same structure: one valid bit followed by 7 bits codifying the corresponding page frame number. At a given moment, the contents of the memory are:

**Note:** bytes from each page are read left (byte 0) to right (byte 31).

page 0:	1a 16 1a 10 17 09 06 11 16 1e 12 0c 07 10 1a 0c 15 06 1d 17 10 00 12 16 18 1c 00 17 0d 08 1e 02
page 1:	0c 08 14 15 18 1c 14 1b 01 16 00 10 08 04 1e 1d 09 03 1a 1d 0c 17 1d 08 0a 0b 05 0d 17 1d 03 13
page 2:	00 00
page 3:	00 00
page 4:	00 00
page 5:	7f b1 7f 7f 7f 7f 7f
page 6:	7f 7f 7f bb 7f 7f 7f e1 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f fa 7f 7f 7f 7f 7f 7f 7f da ea 7f
page 7:	17 00 17 07 1e 0f 1e 09 1d 09 02 0f 0d 0b 03 1b 06 0d 0c 01 14 06 0a 10 0d 0f 1e 0f 1d 1a 13 03
page 8:	00 00
page 9:	00 00
page 10:	15 02 0b 1d 13 00 08 15 0a 0f 18 11 18 12 18 08 15 12 0e 17 0f 1b 19 17 11 05 04 09 11 1a 11
page 11:	0f 05 15 0d 05 1b 0c 08 16 1c 11 16 02 04 0f 15 09 07 08 02 0e 14 13 0a 0d 04 09 0e 17 16 1c 01
page 12:	0e 10 1e 04 14 0b 0f 06 14 07 0e 01 1e 0f 0e 16 0c 1b 00 19 0e 19 1d 1e 05 15 03 04 02 09 00 1a
page 13:	0c 1b 16 0f 14 11 17 1a 0f 1b 06 01 18 0a 0d 02 0d 02 03 0b 12 07 0c 07 07 07 0b 10 0c 19 11 14
page 14:	19 05 15 03 0c 09 1e 01 1b 10 02 1e 01 0d 02 16 03 06 16 0a 1c 0a 16 01 0e 00 0a 09 16 0d 15 01
page 15:	7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f e4 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f d9 7f 7f
page 16:	7f 7f 7f 7f aa 7f 7f 96 7f
page 17:	16 18 0d 0a 0c 00 15 0a 1a 0c 17 14 03 17 05 00 14 09 1e 00 09 04 15 12 1e 1a 00 1b 19 1b 0c 16
page 18:	16 12 08 1a 01 13 0f 19 03 1a 0a 0f 06 02 0d 05 0d 05 02 0c 0c 0a 03 15 19 18 0c 05 02 07 0f 0a
page 19:	11 01 15 11 13 03 09 05 1e 18 01 12 19 16 05 1a 18 17 08 11 11 15 17 0f 0f 1e 14 04 01 0c 07 16
page 20:	00 00
page 21:	00 00
page 22:	0c 0c 1c 14 15 02 1c 15 08 1a 14 11 15 1c 12 09 1a 06 09 16 0b 12 06 0a 1b 06 0a 1a 18 13 10 05
page 23:	7f b7 7f
page 24:	7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f c6 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f a4 7f 7f 7f 7f 7f
page 25:	00 00
page 26:	17 15 02 09 0c 0f 0f 0e 08 17 01 11 1c 06 0e 1d 0c 15 15 0a 12 10 0c 1a 0c 1a 12 0a 1a 0b 1e 03
page 27:	7f db 7f 7f 7f 7f 7f ee 7f 7f 7f 7f
page 28:	18 12 00 00 07 1b 19 1b 00 1d 04 0c 17 06 02 06 06 0b 1c 15 02 01 08 08 06 0f 18 17 01 1d 19 0b
page 29:	1b 1c 07 02 0a 13 0a 18 1b 12 00 04 03 1d 01 0d 02 1b 13 0b 17 08 0f 15 14 1e 1a 1a 17 01 02 06
page 30:	1e 1e 09 19 00 04 05 05 0e 07 1e 16 0c 17 03 14 01 1a 06 1a 18 18 05 09 19 06 0e 05 17 08 0e 00
page 31:	00 00
page 32:	00 00
page 33:	7f 7f 7f 7f 87 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f c8 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
page 34:	7f 7f 7f 7f 7f 7f 7f 7f 7f 7f eb 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
page 35:	08 07 1e 06 10 0f 16 01 1e 0d 1a 05 09 19 1d 10 05 18 10 06 07 01 05 0b 15 0f 10 1c 0c 18 0c 1e
page 36:	05 11 0c 0d 06 14 0e 1e 14 12 0c 0f 14 0e 1d 11 07 14 1a 1d 01 18 00 1b 15 0b 0a 01 06 1a 00 0d
page 37:	1d 1a 03 0e 0c 1b 1a 00 1e 1c 18 15 0e 0b 09 18 03 00 0f 04 0e 0f 1b 1a 0d 18 00 0a 07 0f 1b 1e
page 38:	7f 7f 7f bf 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f fb 7f 7f 7f 7f 7f 7f b3 7f 7f 7f 7f fd 7f 7f 7f
page 39:	7f 7f 9e 7f cc 7f 7f 7f 7f
page 40:	7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f a5 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 91 7f 7f 7f 7f 7f 7f 7f
page 41:	14 07 1d 07 0e 02 05 11 01 0e 01 1e 0e 0c 02 14 1b 02 1d 08 11 0d 11 17 1e 13 14 03 00 19 18 0b
page 42:	0e 03 09 09 17 1c 05 1c 0f 0d 01 16 17 14 19 17 0f 06 15 18 17 04 02 1d 14 08 01 1a 04 1c 15 03
page 43:	00 00
page 44:	7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 9a 7f 7f 7f 7f 7f 7f 7f 7f e6 7f 7f 7f 7f 7f 7f 7f 7f
page 45:	7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f b9 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
page 46:	00 00
page 47:	1d 17 10 19 09 05 1b 1b 1a 0c 1a 0f 1e 1b 18 03 0a 06 0a 07 0f 0f 11 05 1e 11 0f 05 06 1a 17 19
page 48:	00 00
page 49:	02 19 1e 1a 19 05 0f 11 08 0c 04 0a 19 1d 1e 0b 12 04 18 06 01 13 07 1b 03 08 11 09 1a 13 04 12
page 50:	00 00
page 51:	04 0d 16 02 0e 0c 1c 04 1a 11 0f 1b 0e 18 00 16 1b 07 11 02 12 0a 08 1d 09 03 0c 0e 03 0c 08 16
page 52:	00 00
page 53:	0a 0e 19 15 05 1c 11 18 02 07 1a 12 12 0b 11 19 11 16 07 0b 01 04 11 1c 07 0e 0e 1e
page 54:	00 00
page 55:	19 0d 07 02 04 06 1d 16 0d 1d 02 1e 0d 0c 1b 0a 0f 06 17 11 0c 1c 08 18 12 13 11 0c 00 07 0f 09
page 56:	00 00
page 57:	0a 0e 18 1d 1e 13 0f 0a 00 02 00 1b 07 0e 17 02 13 06 1c 1a 0c 11 1e 05 03 1c 0a 17 1c 0e 14 1e
page 58:	00 00
page 59:	19 00 14 08 1b 07 1d 06 1b 13 13 00 12 04 0e 04 12 1c 15 19 04 1b 1e 1b 14 19 18 00 0e 06 1c 0a
page 60:	7f 7f
page 61:	00 00
page 62:	00 15 0d 0e 0d 13 11 05 09 16 15 18 1c 08 10 0b 0f 06 03 03 1e 05 11 17 1e 16 1a 08 0d 11 00 10
page 63:	0b 02 0e 1e 18 1a 1a 13 0d 0f 10 04 03 08 11 03 18 0e 0f 0c 02 19 11 0e 01 0d 0d 11 12 1b 07 07
page 64:	00 00
page 65:	19 06 10 06 01 05 0e 16 0b 0a 1c 02 18 01 1e 0d 02 09 00 08 06 1b 16 07 0a 13 18 14 18 04 0e 18
page 66:	7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 92 7f 7f 7f 7f af 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
page 67:	00 00
page 68:	00 00
page 69:	0c 0e 11 17 04 01 1e 17 12 01 03 14 0d 09 1c 04 0b 05 14 1c 13 0e 0f 0c 07 18 1a 17 18 1e 0a 0c
page 70:	09 09 14 07 13 1b 1a 09 0e 0f 08 0a 1e 00 04 14 02 09 18 1c 0b 06 1b 13 0f 0a 0a 09 17 0e 06 1b
page 71:	00 00
page 72:	01 10 16 10 11 18 11 07 0d 0e 00 0f 0e 19 03 13 1b 05 02 0e 0a 08 11 19 18 17 13 1a 1a 16 0a 0a
page 73:	00 00
page 74:	04 10 1c 04 1e 13 16 15 17 06 12 07 03 09 1c 1d 0f 13 05 08 14 08 17 19 0d 0c 05 07 19 08 02 16
page 75:	7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 80 7f 7f 7f 7f c5 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
page 76:	04 06 0e 06 1a 1e 15 15 0e 17 03 1b 1a 10 1e 06 05 10 1c 19 1d 18 02 02 19 01 0a 17 00 11 13 18
page 77:	00 00
page 78:	00 00
page 79:	04 01 0a 0b 14 11 13 13 13 1e 0e 01 05 06 16 08 1e 17 19 0d 11 12 1a 08 12 13 1e 15 19 18 0b 16
page 80:	0b 16 11 00 14 0c 12 1d 08 01 1c 11 0b 17 02 06 01 02 0c 19 14 1e 04 17 03 14 0e 03 07 14 07 0b
page 81:	00 00
page 82:	00 00
page 83:	7f 7f 7f 7f f0 7f
page 84:	02 0d 0f 00 0f 1c 04 0b 06 10 16 14 04 16 13 1d 13 02 15 0a 01 18 11 0d 11 0e 18 1e 0a 16 1c 0b
page 85:	7f be 7f 7f 7f 7f 7f 7f 7f 7f 7f
page 86:	1d 14 0f 0a 16 00 1e 04 0d 00 0e 09 03 15 1b 0f 06 0d 05 1b 11 0e 18 0a 16 0a 0b 0c 10 07 08 00
page 87:	7f 7f
page 88:	00 00
page 89:	0e 08 06 07 1b 10 07 19 12 1b 0e 0f 1d 0d 00 02 05 1d 0b 12 17 13 18 02 00 0b 02 07 17 0b 17 03
page 90:	15 04 0a 11 19 1c 10 1e 09 02 16 02 1b 10 0d 14 01 0e 1b 04 0e 16 07 02 04 08 0f 1c 0b 10 18 12
page 91:	12 13 07 04 17 10 0d 0e 18 19 0c 17 00 1b 00 1e 1e 12 1b 14 02 15 1e 16 06 0d 1a 18 06 19 0a 00

