OPERATING SYSTEMS I

First and Last Name: ………………………………………………………………………………………………………

Year of Study: …………………

31st of January, 2021

No points granted by default; minimum points required to pass exam: ***50p***

EXERCISE 1: PROCESS SCHEDULING (25p)

1. A system with 5 processes needs to be scheduled according to the algorithms below. Using the information below, fill in the missing parts for all algorithms presented and answer the questions below.

			RR		SJF		SRTN		FCFS	
	Arrival	**Burst**	T_W	T_R	T_W	T_R	T_W	T_R	T_W	T_R
P1	2	8								
P2	0	13								
P3	1	22								
P4	2	11								
P5	8	7								

a) Find out the waiting time and the turnaround time for each of the processes through the GANTT diagram, for each of the algorithms presented. Wherever necessary, the quantum value (*q*) is the optimal one. (16p)

b) Which algorithm offers the best performance, taking into account that *q* is optimal, based on the average values for the waiting time and running time? Explain briefly the performance gains. (5p)

c) Which algorithm offers the lowest number of context switches in the situation above, considering that *q* is optimal? (1p)

d) What would happen if *q* = 11 in terms of response times and context switches, for the Round-Robin algorithm? Explain briefly which situation is more favorable and why. (1p)

e) Write down the process queue when scheduling occurs at the moment of time 30, in the SRTN algorithm. (1p)

In the SJF algorithm, besides the first process to execute, which process has the smallest response time? (1p)

EXERCISE 2: PROCESS SCHEDULING (50p)

2. A system with 10 processes needs to be scheduled according to the algorithms below. Using the information below, fill in the missing parts for all algorithms presented and answer the questions below.

			RR		SJF		SRTN		FCFS	
	Arrival	**Burst**	T_W	T_R	T_W	T_R	T_W	T_R	T_W	T_R
P1	29	15								
P2	26	22								
P3	22	14								
P4	13	35								
P5	10	37								
P6	0	3								
P7	5	6								
P8	5	21								
P9	0	31								
P10	5	13								

f) Find out the waiting time and the turnaround time for each of the processes through the GANTT diagram, for each of the algorithms presented. Wherever necessary, the quantum value (*q*) is the optimal one. (30p)

g) Which algorithm offers the best performance, taking into account that *q* is optimal, based on the average values for the waiting time and running time? Explain briefly the performance gains. (10p)

h) Which algorithm offers the lowest number of context switches in the situation above, considering that *q* is optimal? (2.5p)

i) What would happen if *q* = 40 in terms of response times and context switches, for the Round-Robin algorithm? Explain briefly which situation is more favorable and why. (2.5p)

j) Write down the process queue when scheduling occurs at the moment of time 30, in the SRTN algorithm. (2.5p)

k) In the SJF algorithm, besides the first process to execute, which process has the smallest response time? (2.5p)

EXERCISE 3: DEADLOCK (30p)

3. In a system with a single type of resources, there are 8 processes with the following maximal requirements:

Process	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**
MAX	15	60	65	35	30	45	30	30

a) Specify the minimal value for the total number of available resources, so that the state of the system is considered safe, in the following scenarios:

a1) an initial allocation of resources: (5, 10, 0, 5, 15, 0, 0, 10) (10p)
a2) an additional allocation of resources: (10, 0, 0, 5, 0, 10, 10, 10) (10p)

b) Considering an initial allocation of resources of (5, 10, 0, 15, 5, 0, 0, 10) and a total number of resources equal to the one determined at a1) earlier, answer the following:

b1) give an example of a request which would trigger a deadlock in the system (5p)
b2) two processes request an additional 10 resources, with the state of the system still being safe. One of these processes is **P5**. Find the other process and the maximum number of resources it accessed. (5p)

EXERCISE 4: MEMORY MANAGEMENT (30p)

4. A system using a 32-bit Von Neumann architecture has a page size of 2,048 bytes and 10 KB of RAM memory. Access to the pages of the system is happening in the following order: 4, 4, 2, 1, 6, 7, 3, 3, 1, 5, 1, 6, 5, 1, 5, 6, 6, 5, 5, 5, 0, 1, 3, 4, 1, 2, 2, 1, 0, 7

a) What is the total number of virtual pages and page frames in this system? (5p)
b) How many page faults are issued when using the clock, FIFO and optimal algorithms for the accesses above? (15p)
c) Describe the memory mapping of the pages at moment 15 in the scenario above for all three algorithms mentioned. (2.5p)
d) Considering that at moment 7 you are accessing the virtual address 0xFF00, what is its corresponding page frame? (7.5p)

EXERCISE 5: CHALLENGES & QUESTIONS (65p)

5. Indicate whether each of the statements below is true or false, and explain your answer *briefly* (20p):

a) You cannot have 2 processes running at the same time on a CPU with a single core in an interactive system. (5p)
b) A process can voluntarily force pre-emption in the operating system, to allow other processes to run. (2.5p)
c) Setting a thread's priority to a higher level will guarantee that thread's execution in the next scheduler execution cycle. (2.5p)
d) Virtual addresses are the same as physical addresses, main difference being one is virtual memory, the other physical memory. (2.5p)
e) Page faults happen more often on systems with a low amount of external memory. (2.5p)
f) On multi-core systems, it's mandatory to protect data writes to memory with critical regions to avoid data corruption. (5p)

Answer the following questions *briefly* (45p):
g) Suppose a friend told you: "When designing an operating system for mobile devices (e.g. PlayStation/Xbox), you need to use smaller page sizes and more states for processes". Explain if your friend is right, wrong or both. (5p)
h) The following code snippets modify two variables ***N*** and ***M*** (which are initialized with the values N=1, M=5) in 3 different threads, on a system with 1 CPU, 2 cores and 4 threads:

Thread 0	*Thread 1*	*Thread 2*
`M -= 5;`	`M += 2*N;`	`N /= N*2;`

h.1) Compute the expected result for the variables ***N*** and ***M*** after all threads finish execution in an interactive system, assuming assignment instructions are atomic instructions. (15p)
h.2) Assuming that thread 0 always executes last, and thread 1 has a higher priority than thread 2, compute the value of ***N*** and ***M*** after the execution of the threads complete in an interactive system. (15p)

i) Suppose a friend told you: "A mobile operating system allocates computationally-intensive processes to a higher priority than I/O-bound processes." Explain whether your friend is right, wrong, or both. (5p)
j) What would happen if processes blocked in an I/O event would go directly to the "run" state, when the I/O event happens? (2.5p)
k) Can a thread's change of state (e.g. from "running" to "blocked") influence the state of its associated process? (2.5p)