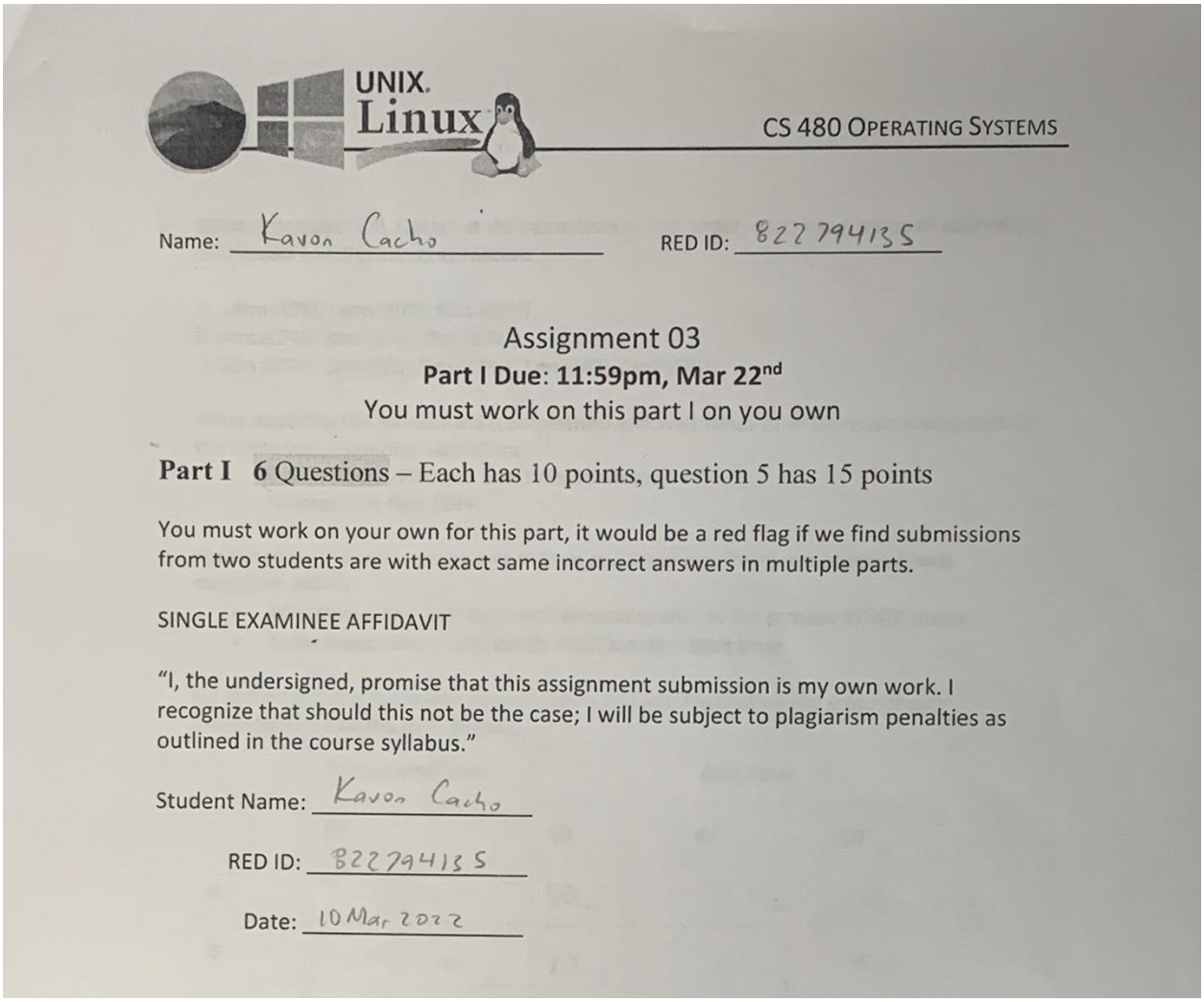
Kavon Cacho
822794135

Affidavit



UNIX. Linux

CS 480 OPERATING SYSTEMS

Name: Kavon Cacho RED ID: 822794135

Assignment 03

Part I Due: 11:59pm, Mar 22nd

You must work on this part I on you own

Part I 6 Questions – Each has 10 points, question 5 has 15 points

You must work on your own for this part, it would be a red flag if we find submissions from two students are with exact same incorrect answers in multiple parts.

SINGLE EXAMINEE AFFIDAVIT

"I, the undersigned, promise that this assignment submission is my own work. I recognize that should this not be the case; I will be subject to plagiarism penalties as outlined in the course syllabus."

Student Name: Kavon Cacho

RED ID: 822794135

Date: 10 Mar 2022

Kavon Cacho
822794135

Question 1

1. Three processes A, B, C start at the same time in that order, each has a series of alternating CPU bursts and I/O bursts as follows:

   A: 14ms (CPU), 3ms (I/O), 6ms (CPU)
   B: 4ms (CPU), 8ms (I/O), 7ms (CPU)
   C: 2ms (CPU), 1ms(I/O), 3ms (CPU), 1ms (I/O), 1ms (CPU)

   What would be the **turnaround (completion) and wait times** of all processes using each of the following scheduling algorithms?
   Round Robin (RR) with **5ms** time quantum,
   Shortest Job First (SJF).

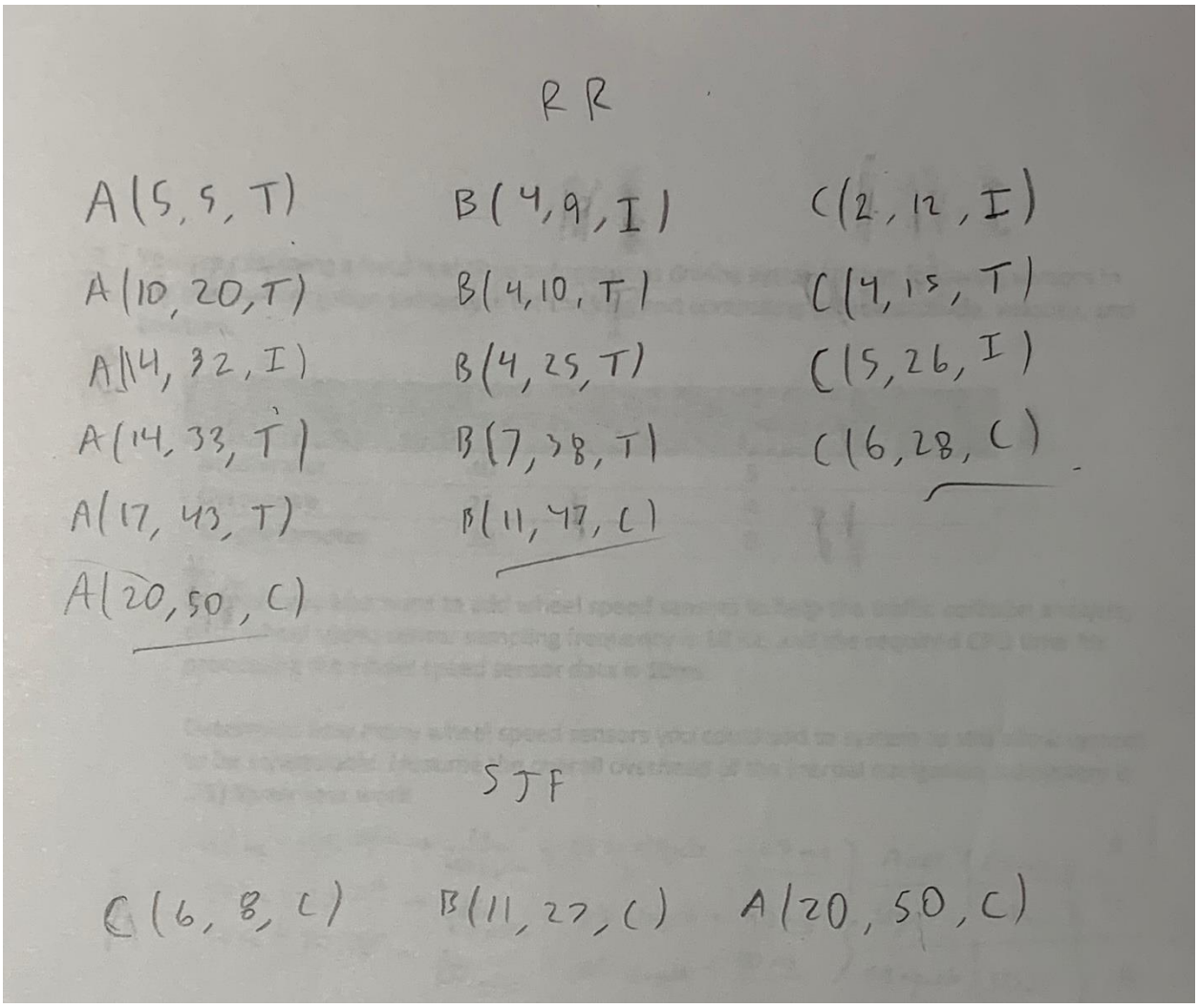
   For simplicity, assume context switch time and other overheads are comparatively negligible. **NOTE:**
   - **Wait time** is the wait time each process spends in the process READY queue.
   - **Turnaround time = CPU bursts + I/O bursts + Wait time**

   Fill in the following table with your answers, and you **must show** the steps of your calculation (follow the hints below).

	Turnaround Time			Wait Time	
	RR	SJF		RR	SJF
A	50 ms	50 ms		22 ms	27 ms
B	47 ms	27 ms		23 ms	8 ms
C	28 ms	8 ms		20 ms	0 ms
Average	41.66 ms	28.33 ms		21.66 ms	11.66 ms

   Hint: Use notation p (e, s, r) to track CPU execution of a process over the time:
   e - total CPU burst time that has been executed for a particular process,
   s - total system time passed from the beginning of executing first process,
   r - the reason why the process is stopped along the execution (either suspended or completed) due to:
   - t - time quantum expired during RR (only applicable to RR),
   - i - process initiated an I/O,
   - c - process completed execution
   - For example: A(10, 16, t) means total CPU burst time that has been executed for process A is 10ms, total system time passed is 16ms since the beginning of executing the first CPU burst, and process A is suspended due to expiration of time quantum.

Kavon Cacho
822794135



Kavon Cacho
822794135

Question 2

2. You are designing a hard real-time autonomous driving system. It has following sensors in the inertial navigation subsystem for tracking and controlling vehicle attitude, velocity, and position:

Sensors	Sampling Frequency (Hz)	CPU time required (ms)
Accelerator	40	5
Gyroscope	25	4
Magnetometer	20	8

Suppose you also want to add wheel speed sensors to help the traffic collision analysis, each wheel speed sensor sampling frequency is **10** Hz, and the required CPU time for processing the wheel speed sensor data is **10**ms.

Determine how many wheel speed sensors you could add to system to still allow system to be schedulable. (Assume the overall overhead of the inertial navigation subsystem is .25) **Show** your work.

$40\,hz = 40\,\frac{cycle}{s} \rightarrow \frac{1s}{40\,cycles} = .025\ s/cycle = 25ms$

$25\,hz = 25\,\frac{cycle}{s} \rightarrow \frac{1s}{25\,cycles} = .04\ s/cycle = 40ms$

$20hz = 20\,\frac{cycle}{s} \rightarrow \frac{1s}{20\,cycles} = .05\ s/cycle = 50ms$

Accel	25ms	5
Gyro	40ms	4
Magnet	50ms	8

$\frac{5}{25} + \frac{4}{40} + \frac{8}{50} + .25 = .71 \leq 1$ schedulable w/o new sensors ✓

Wheel speed sensors: $10hz = 10\,\frac{cycle}{s} \rightarrow \frac{1s}{10cycles} = .10\ s/cycle = 100ms$

$\frac{C_i}{P_i} = \frac{10}{100} = .1$

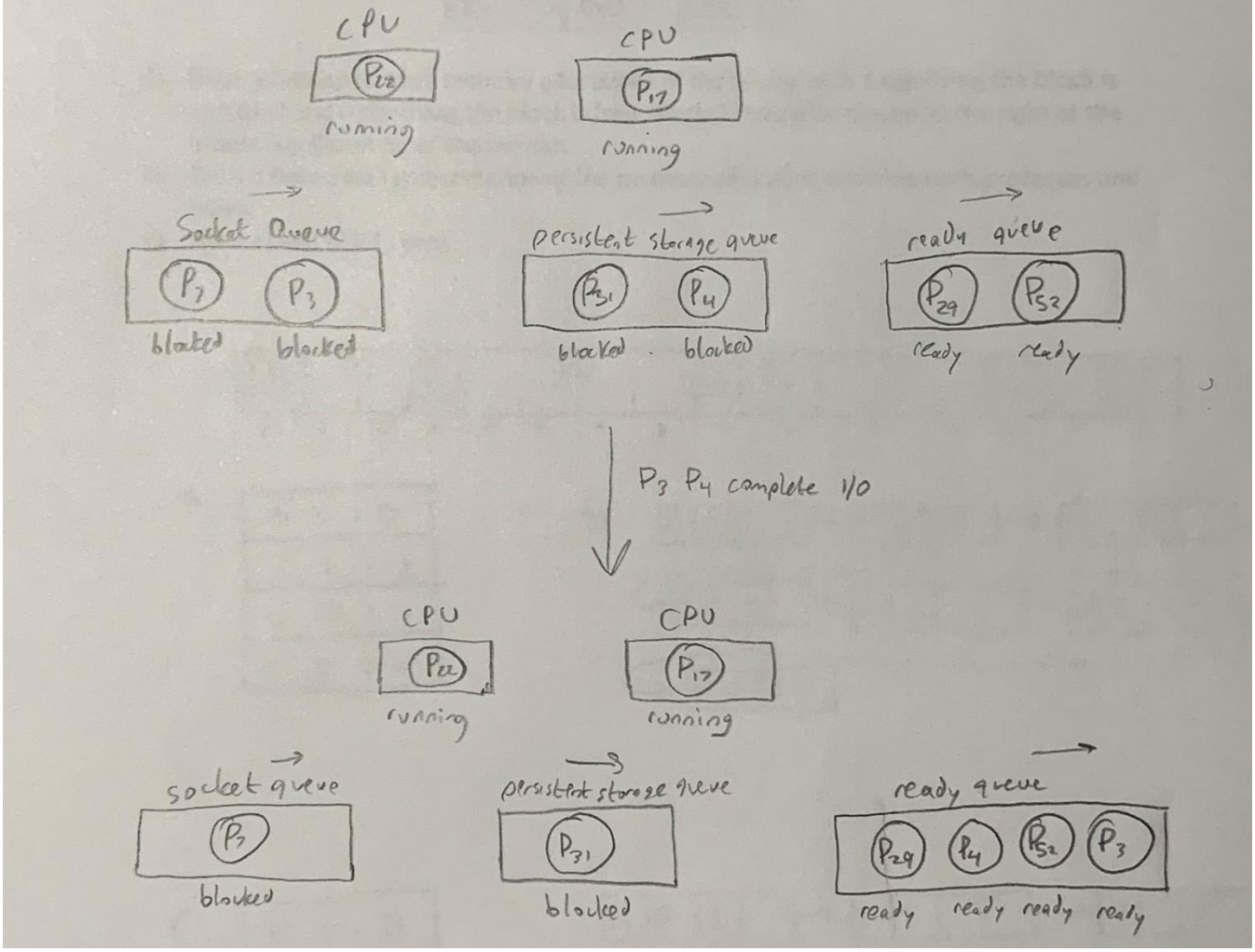
$\frac{5}{25} + \frac{4}{40} + \frac{8}{50} + \frac{10}{100} + \frac{10}{100} + .25 = .91 \leq 1$

2 wheel speed sensors max

Kavon Cacho
822794135

Question 3

3. Recall question 4 prompt from assignment 2, we now apply priority scheduling to the system. The higher the priority number is, the higher the process's priority. Processes P22 (with priority 6) and P17 (with priority 7) are executing. P7 (with priority 6) and P3 (with priority 7, first in the queue) are waiting for a socket write to complete. P4 (with priority 5, first in the queue) and P31 (with priority 4) are waiting on persistent storage access. P52 (with priority 6, first in the queue) and P29 (with priority 4) are awaiting to be scheduled on the CPU.
Suppose P3 and P4 both complete I/O operations at the same time, what would the process queueing diagram for these processes become? (Note each type of I/O has its own waiting queue)



Kavon Cacho
822794135

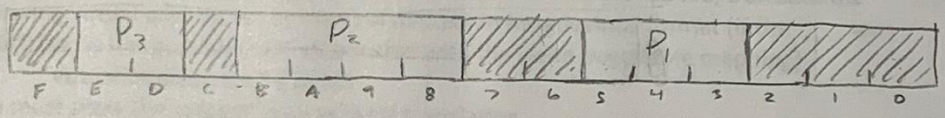
Question 4

4. Suppose a small system has a physical memory size of 16KB, which is divided into 16 blocks (each block has 1KB) for applying a contiguous memory allocation scheme. The blocks are numbered 0x0 through 0xF.

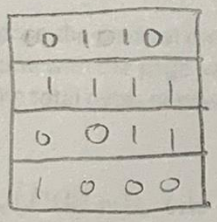
Three processes are loaded in memory for execution:

Process	Start block	End block
P_1	0x3	0x5
P_2	0x8	0xB
P_3	0xD	0xE

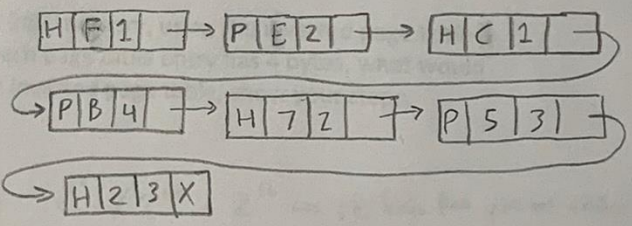
a) Draw a bitmap to track memory allocations of the blocks with 1 signifying the block is occupied and 0 signifying the block is free. Block 0 should be placed to the right at the lowest significant bit of the bitmap.
b) Draw a linked list representation of the memory allocation showing both processes and holes.
c) Redo a) and b) if P_2 exits.



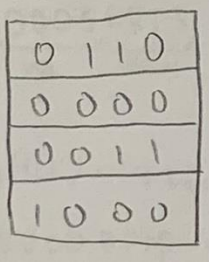
a.

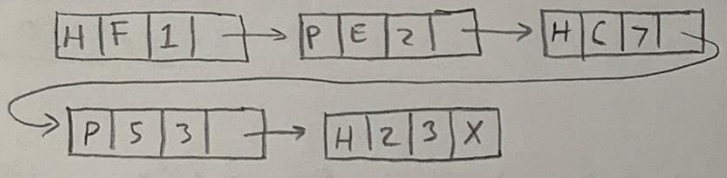


b.



c.





Kavon Cacho
822794135

Question 5

5. Consider a 32-bit virtual memory space. Suppose the page size is set to 64KB, and part of the page table entries is given as below.

	Page Table	
Virtual Page Number	Valid	Physical Frame Number
0x0	1	0xE
0x1	0	0x2
0x2	1	0x20B
0x3	1	0xA2
0x4	1	0x6
0x5	0	0x30
0x6	1	0x725

1) Based on the above information, use binary arithmetic to translate the following virtual addresses to their corresponding physical addresses in 32-bit hexadecimal format.
   Virtual Address: 0x00000A96
   Virtual Address: 0x00036813
   You must show the steps of your address translation.

2) If the page size is set to 16KB, using the same page table entries above, translate the following virtual address to the physical address in hexadecimal format (if the translation is not possible, say not possible with the given page table mapping)
   Virtual Address: 0x00008715
   You must show the steps of your address translation.

3) Suppose the physical memory space has 2GB memory, using the inverted page table scheme with the page size as 8KB, and each page table entry has 4 bytes, what would be the total bytes needed for storing the inverted page table, show your steps.

1. $\log_2(64K) = 16$ bits for offset $2^{32}/2^{16} = 2^{16} \rightarrow 16$ bits for virtual address

$0x\underbrace{0000}_{VA}\underbrace{0A96}_{offset} \rightarrow$ **0x000E0A96**

$0x\underbrace{0003}_{VA}\underbrace{6813}_{offset} \rightarrow$ **0x00A26813**

2. $\log_2(16K) = 14$ bits for offset $2^{32}/2^{14} = 2^{18} \rightarrow 18$ bits for virtual address

0x00008715 → 0000 0000 0000 0000 1000 0111 0001 0101

VA 7 | offset

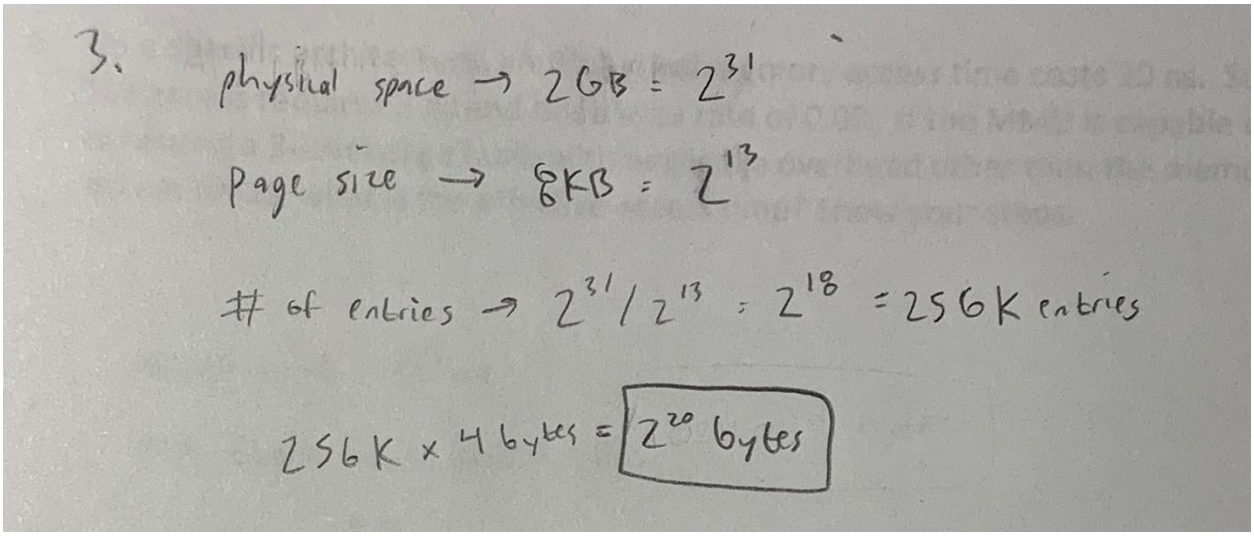
0x0020B → 0000 0010 0000 0000 1011

→ 1011 = 1000

0000 0010 0000 0000 0000 1000 0111 0001 0101

= 0x020008715

Kavon Cacho
822794135



Question 6