User Id: ________

CSE 303: Quiz #4

Due November 28th, 2022 at 11:59 PM

The quiz has THREE questions. Please submit your answer on CourseSite as a pdf whose name is exactly your user Id and the “pdf” extension (e.g., abc123.pdf) before the deadline.

Question 1: We studied four scheduling algorithms in lectures: FIFO, SJF, RR, MFQ.

(a) Which one of them is closer to the way you schedule your homework assignments? One valid answer is “neither of them” -- in this case explain what is the scheduling algorithm you follow.

(b) Use the terms we discussed in Chapter 7 (response time, fairness, throughput, starvation, …) to argue why or why not you should stick with the way you schedule your homework assignments.

Give a detailed answer, but keep your entire response to a single side of an 8.5x11 page.

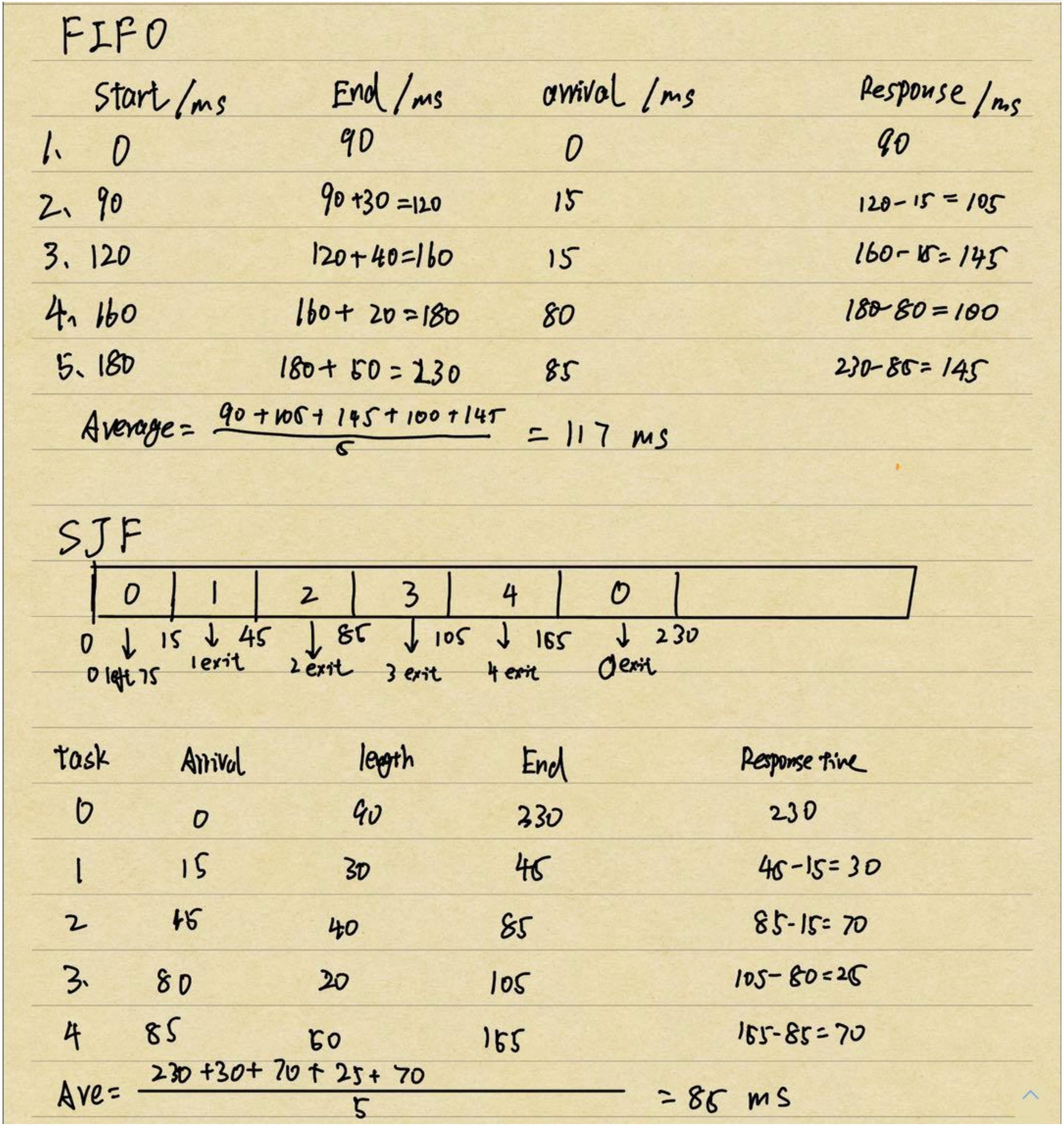
a).Normally, for weekly assignments, which are usually due on Sunday. I would follow the SJF algorithm. In this case, I will normally do the shortest one first, for the reason that longer assignments are generally harder than short one, and some shorter assignments, like 10 MCQ will be easier to receive higher grades. Therefore, I will firstly ensure the completion of the shorter one to receive higher grades. Then after I finished all the easy and short assignments, I will move on to the longer and harder assignments. If I do the longer one first, which I do not have 100% of confidence of getting good grade, and get stuck on longer one, I may have no time to do the easy and short one and lose the grades I could have gotten. Also, I think the algorithm I followed is the not-preemptive version, which means if I find another shorter assignment while I am doing the one that is longer, I generally will not switch to the shorter one.

b) The reason why I should stick with this way is the one above, which I think is the advantage of the SJF: optimal average response time. However, the biggest problem is the possible starvation. If I have too much shorter assignments, after I finish each one of them, I will probably have no time to do or complete the longer one, which is an unfairness by default of SJF.

Question 2: Given the following mix of tasks, task lengths, and arrival times, compute the completion and response time for each task, along with the average response time for the FIFO, RR, and SJF algorithms. Assume a time slice of 20 milliseconds and that all times are in milliseconds.

Task	Length	Arrival Time
0	90	0
1	30	15
2	40	15
3	20	80
4	50	85

User Id: ________



FIFO

	Start /ms	End /ms	arrival /ms	Response /ms
1、	0	90	0	90
2、	90	90+30=120	15	120-15=105
3、	120	120+40=160	15	160-15=145
4、	160	160+20=180	80	180-80=100
5、	180	180+50=230	85	230-85=145

$$\text{Average} = \frac{90+105+145+100+145}{5} = 117 \text{ ms}$$

SJF

0	1	2	3	4	0	

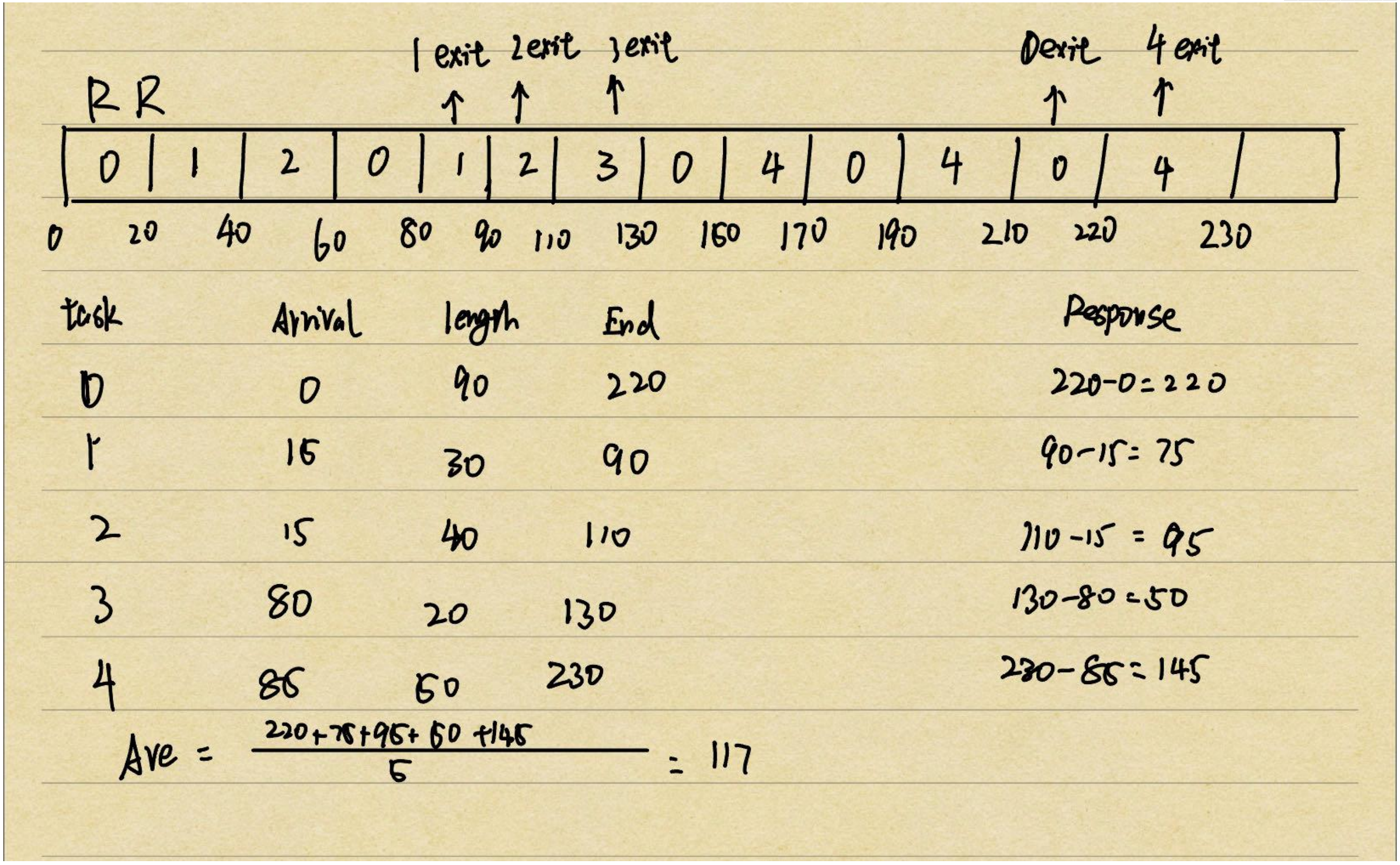
0 ↓ 15 ↓ 45 ↓ 85 ↓ 105 ↓ 165 ↓ 230

0 left 75 | 1 exit | 2 exit | 3 exit | 4 exit | 0 exit

task	Arrival	legth	End	Response time
0	0	90	230	230
1	15	30	45	45-15=30
2	15	40	85	85-15=70
3.	80	20	105	105-80=25
4	85	50	165	165-85=70

$$\text{Ave} = \frac{230+30+70+25+70}{5} = 85 \text{ ms}$$

User Id: ________


RR
1 exit
2exit
3exit
0exit
4 exit
0
1
2
0
1
2
3
0
4
0
4
0
4
0
20
40
60
80
90
110
130
160
170
190
210
220
230

task	Arrival	length	End	Response
0	0	90	220	220-0=220
1	15	30	90	90-15=75
2	15	40	110	110-15 = 95
3	80	20	130	130-80=50
4	85	50	230	230-85=145

$$\text{Ave} = \frac{220+75+95+50+145}{5} = 117$$

Question 3: Assume we have a 32-bit virtual address that is split into 20 bits for virtual page number and 12 bits for page offset.

A. What is the maximum number of virtual pages in this system? What is the size of each page?
B. Is it possible to have a different number of physical frames than the number of virtual pages? Why or why not?
C. Is it possible to have a different physical frame size than the virtual page size? Why or why not?
D. Considering a single-level page table is used to translate those virtual pages into physical page frames, and assuming that each page table entry is 4 bytes, how many pages are needed to load this entire page table into memory?

a. number of virtual page = 2^20 pages, size = 2^12 = 4 kb

b. Yes. it is possible to have a different number of physical frames than the number of virtual pages. According to the main idea of virtual memory, it allow applications to access more memory than is physically present on the machine , an illusion of a nearly infinite amount of physical memory. For example, if the system only has 1,000,000 physical frames, then there would be 1,048,576 - 1,000,000 = 48,576 virtual pages that do not have a physical frame. This would mean that some virtual pages would need to be swapped out to disk when they are not being used, which would add some overhead to the system.

c. Yes,it is possible to have a different physical frame size than the virtual page size. For example, if the physical frame size is 2^14, which is 16,384 bytes, then the system would have 1,048,576 / 16,384 = 64 physical frames for every 1,048,576 virtual pages. This would mean that the physical address space would be 64 times smaller than the virtual address space, which could lead to some wasted space if the pages are not fully utilized.

d. 2^20 / 2^12 = 246 pages