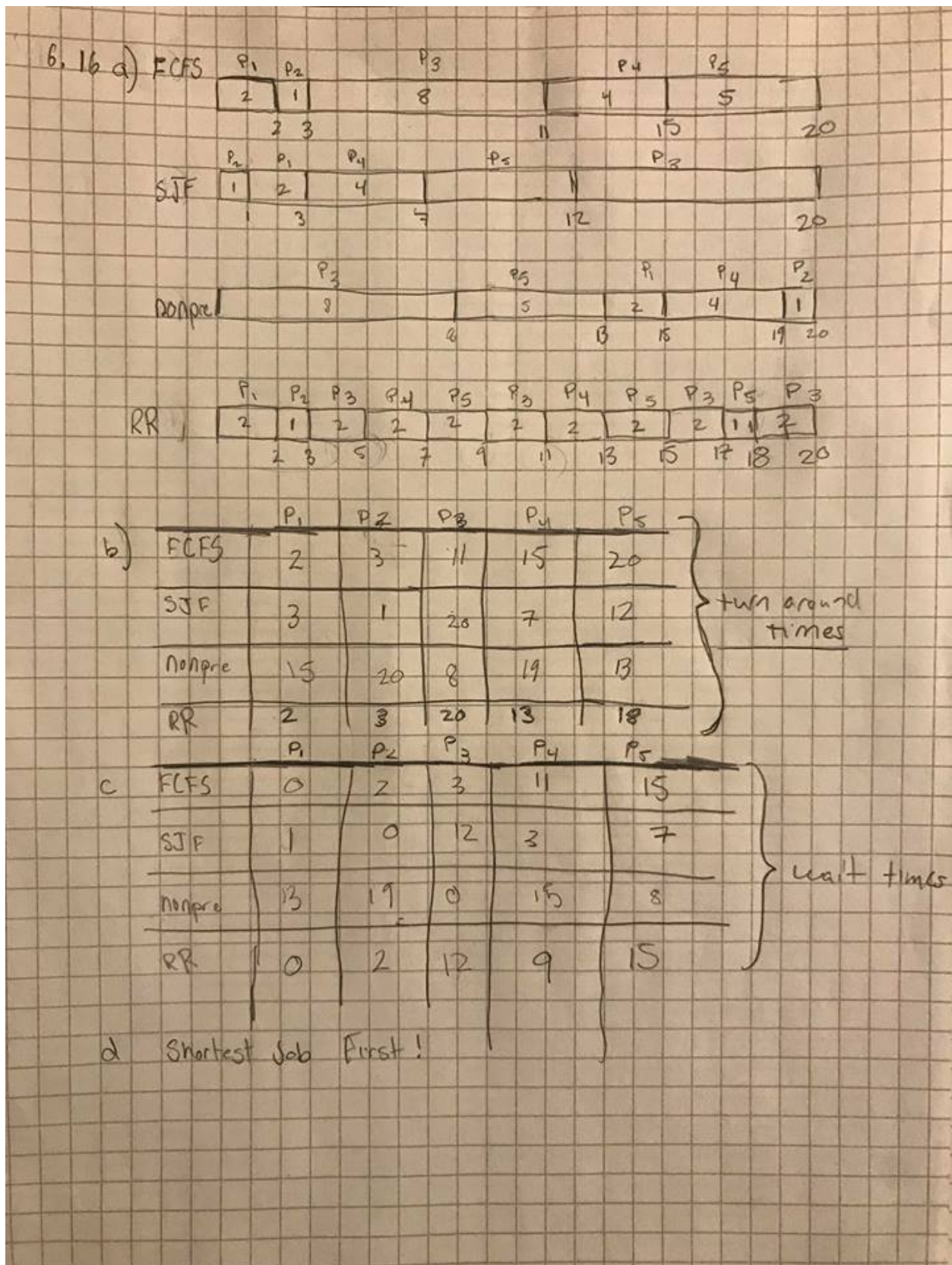**5.10: Explain why implementing synchronization primitives by disabling interrupts is not appropriate in a single-processor system if the synchronization primitives are to be used in user-level programs.**

By doing this, the single processor system would not be able to potentially leave the process by using context switches. This could potentially cause one process to run perpetually or not letting other processes that are needed to run to not run.

5.11: Explain why interrupts are not appropriate for implementing synchronization primitives in multiprocessor systems.

Interrupts are not appropriate for implementing synchronization because when you disable an interrupt, it prevents the processes from the processor the interrupt was disable for. This ultimately means that the process which disabled interrupts cant ensure the same access to the program state which is needed to achieve synchronization.

6.16:

6.16 a)

FCFS

P1	P2	P3	P4	P5
2	1	8	4	5

2   3   11   15   20

SJF

P2	P1	P4	P5	P3
1	2	4		

1   3   7   12   20

Nonpre

P3	P5	P1	P4	P2
8	5	2	4	1

8   6   13   15   19   20

RR

P1	P2	P3	P4	P5	P3	P4	P5	P3	P5	P3
2	1	2	2	2	2	2	2	2	1	2

2   3   5   7   9   11   13   15   17   18   20

b)

	P1	P2	P3	P4	P5	
FCFS	2	3	11	15	20	turn around times
SJF	3	1	20	7	12	
Nonpre	15	20	8	19	13	
RR	2	3	20	13	18	

c)

	P1	P2	P3	P4	P5	
FCFS	0	2	3	11	15	wait times
SJF	1	0	12	3	7	
Nonpre	13	19	0	15	8	
RR	0	2	12	9	15	

d)

Shortest Job First!

6.27:
   a) 160 and 40
   b) 40
   c) 52