

SYSC 4001 Operating Systems Fall 2025  
 Assignment 2 - L1-34  
 Mark Bowerman 101272081

**Part 1 - Concepts**

- a) Consider the following set of processes. Each process has a single CPU burst and does not perform any I/O.

Process	Arrival Time (ms)	Execution Time (ms)
P1	0	12
P2	5	8
P3	8	3
P4	15	6
P5	20	5

With the help of Gantt charts, draw the execution timeline for the following scheduling algorithms:

i) FCFS



	Completion Time	Turnaround time	Mean Turnaround Time
P1	12	12	
P2	20	15	
P3	23	15	
P4	29	14	
P5	34	14	$(12+15+15+14+14)/5 = 14$

**ii) Round Robin algorithm with a time slice of 4 ms.**

P1	P1	P2	P3	P1	P2	P4	P5	P4	P5
0	4	8	12	15	19	23	27	31	33

	Completion Time	Turnaround time	Mean Turnaround Time
P1	19	19	$(19+18+7+18+14)/5 = 15.2$
P2	23	18	
P3	15	7	
P4	33	18	
P5	34	14	

**iii) Shortest Job First with preemption**

P1	P1	P3	P1	P4	P4	P5	P2
0	5	8	11	15	20	21	26

	Completion Time	Turnaround time	Mean Turnaround Time
P1	15	15	$(15+29+3+6+6)/5=11.8$
P2	34	29	
P3	11	3	
P4	21	6	
P5	26	6	

iv) Multiple queues with feedback (high-priority queue: quantum = 2; mid-priority queue: quantum = 3; low-priority queue: FIFO)

High	P1		P2		P3				P4		P5					
Med		P1		P2		P2	P3			P4		P5				
Low								P1					P1	P2	P4	
	0	2	5	7	8	10	13	14	15	17	20	22	25	31	33	34

	Completion Time	Turnaround time	Mean Turnaround Time
P1	31	31	$(31+28+6+19+5)/5=17.8$
P2	33	28	
P3	14	6	
P4	34	19	
P5	25	5	

- b) Now assume that each process in part a) requests to do an I/O every 2 ms, and the duration of each of these I/O is 0.5 ms. Create new Gantt diagrams considering the I/O operations and repeat all the parts done in part a) using this new input trace.

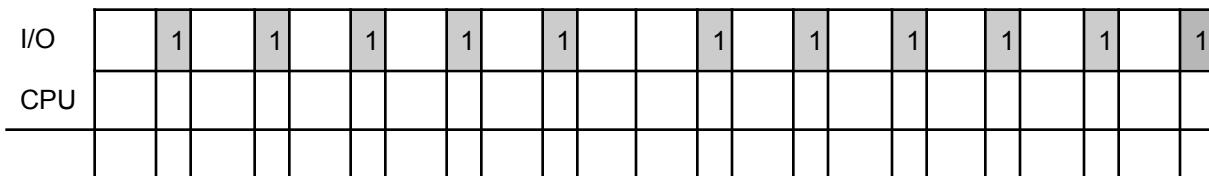
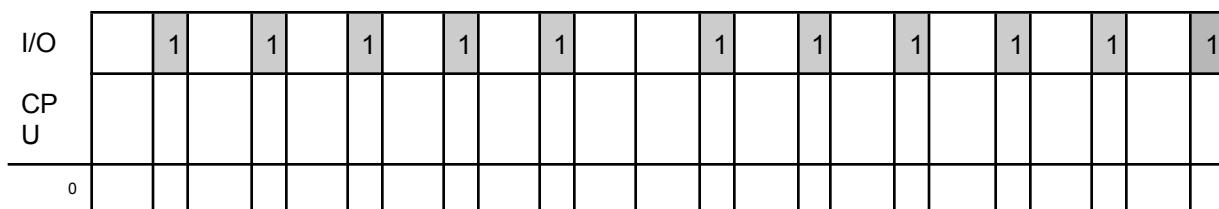
### i) FCFS

I/O		1		1		1		1		1		1		1		1	
CPU	P1		P1		P1		P1		P1		P2		P2		P2		P3
	0	2		4.5		7		9.5		12		14.5		17		19.5	

I/O		1		1		1										
CPU	P4		P4		P4		P5									
	28.5	30.5		33		35.5		38		39.5						

	Completion Time	Turnaround time	Mean Turnaround Time
P1	14.5	14.5	$(14.5+19.5+20.5+20.5+19.5)/5=18.9$
P2	24.5	19.5	
P3	28.5	20.5	
P4	35.5	20.5	
P5	39.5	19.5	

ii) Round Robin with q=4.



	Completion Time	Turnaround time	Mean Turnaround Time
P1			
P2			
P3			
P4			
P5			

### iii) Shortest Job First with Preemption

	Completion Time	Turnaround time	Mean Turnaround Time
P1			
P2			
P3			
P4			
P5			

iv) Multiple queues with feedback (high-priority queue: quantum = 2; mid-priority queue: quantum = 3; low-priority queue: FIFO)

- c) Consider a multiprogrammed system that uses multiple partitions (of variable size) for memory management. A linked list of holes (the “free” list) is maintained by the operating system to keep track of the available memory in the system.

At the start of the exercise, the free list consists of holes with the following sizes (in KB):

The free list is ordered in the sequence given above: the first hole is 85 KB, followed by 340 KB, and so on.

Position	Hole Size	Status
1	85 KB	Free
2	340 KB	Free
3	28 KB	Free
4	195 KB	Free
5	55 KB	Free
6	160 KB	Free
7	75 KB	Free
8	280 KB	Free

When the system is in this state, the following jobs arrive; each of them has different memory requirements, and they arrive in the following order:

With  $t_1 < t_2 < t_3 < t_4 < t_5$

Job No.	Arrival Time	Memory Req.
J1	$t_1$	140 KB
J2	$t_2$	82 KB
J3	$t_3$	275 KB
J4	$t_4$	65 KB
J5	$t_5$	190 KB

1) Determine which free partition will be allocated to each job for the following algorithms:

- (i) First Fit
- (ii) Best Fit
- (iii) Worst Fit

For each algorithm, show the allocation table (which job gets which partition), the remaining free memory after all allocations, the total internal fragmentation and the total external fragmentation.

- (i) First Fit

Position	Hole Size	Status
1	85 KB	J2
2	340 KB	J1
3	28 KB	Free
4	195 KB	J4
5	55 KB	Free
6	160 KB	Free
7	75 KB	Free
8	280 KB	J3

- (ii) Best Fit

Position	Hole Size	Status
1	85 KB	J2
2	340 KB	Free
3	28 KB	Free
4	195 KB	J5
5	55 KB	Free
6	160 KB	J1
7	75 KB	J4
8	280 KB	J3

- (iii) Worst Fit

Position	Hole Size	Status
1	85 KB	Free
2	340 KB	J1
3	28 KB	Free
4	195 KB	J4
5	55 KB	Free
6	160 KB	Free
7	75 KB	Free
8	280 KB	J2

- 2) Based on your calculations, analyze and compare the three algorithms according to memory utilization efficiency and fragmentation. Based on your analysis, justify which algorithm would be most appropriate for a system with frequent small allocations, and a system with mixed workload sizes. Show all calculations step by step; also draw the memory state after each allocation. Calculate fragmentation metrics for comparison and provide detailed justification for your analysis.