Nov 7 2025
SYSC 4001

PART 1

Group 34
Esteban Heidrich - 101267959
RIley Foxton - 101304022

Individual parts

FCFS Explanation:

The FCFS or First Come First Serve algorithm is the simplest of the CPU-scheduling algorithms. It is also non-preemptive. When a program enters the ready queue, it is linked to the end of the queue. Whenever the CPU is free it begins processing the program at the front of the queue

Round Robin Explanation:

The Round Robin (RR) algorithm works by setting a time interval, called a slice or quantum, and letting a process run for that amount of time before scheduling the next process. This is achieved by using a timer to interrupt the process every slice interval. If there is only one process in the ready queue then the timer still interrupts the process so the scheduler can check if there are any other processes there, if not, the same process continues to run. If a process takes less time then the slice to finish executing then the next process runs early.

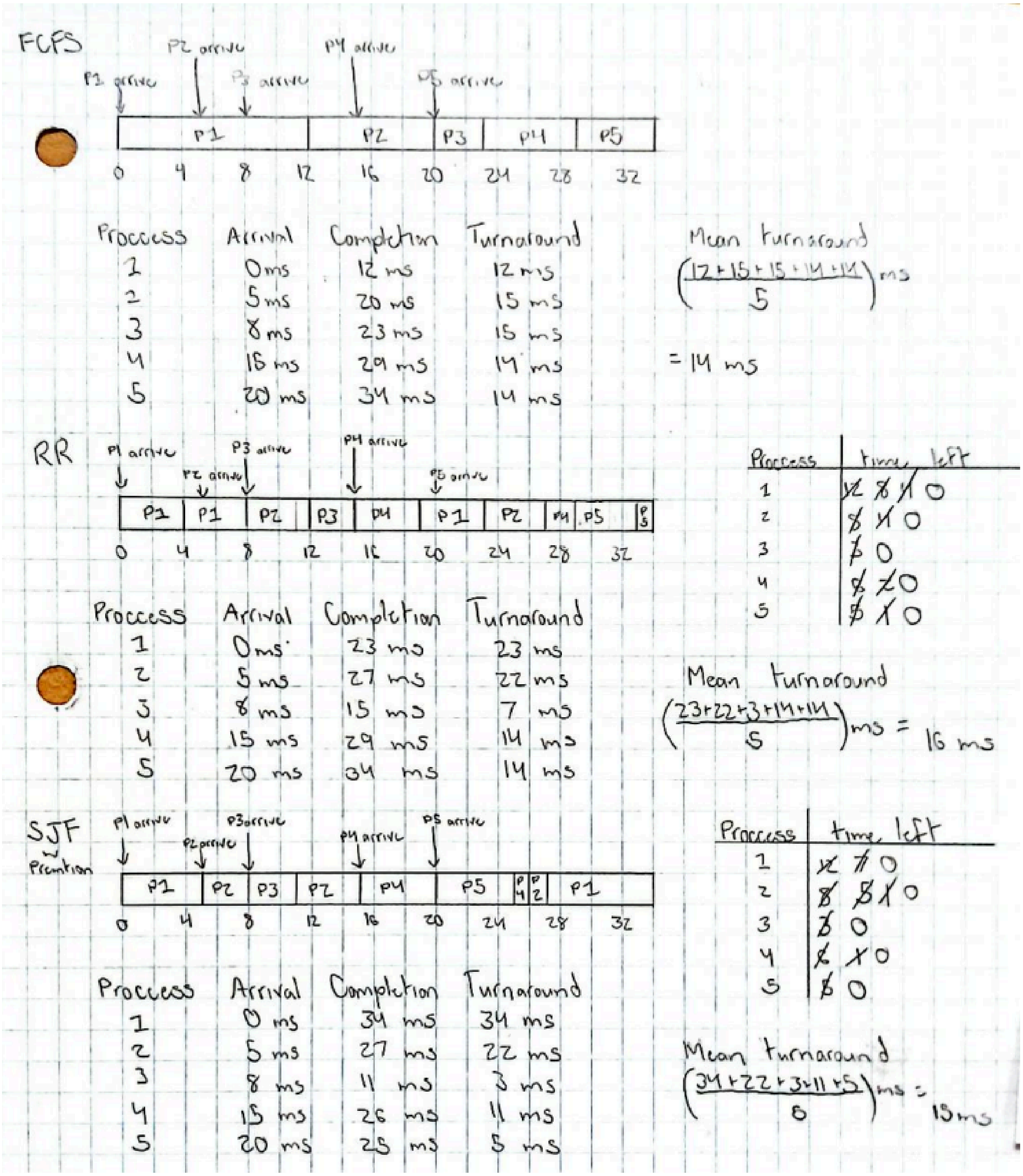
Exec Explanation:

The exec function replaces a process's image with a new one. It however keeps the old process's ID. This is important as this function is often combined with the fork() function to create a process that can communicate with its parent but is unique in functionality

Fork Explanation:

The fork() system call duplicates the process that calls it, so that the new process (the child) is identical to the process that called it (the parent). The parent and the child will function exactly the same way from the point of the fork. However they will have different process identification numbers, memory spaces and PCBs. The parent's PID returned by fork() will be the actual PID of the child in the PCB block and the PID returned by fork() of the child will be 0. If for some reason the fork() call fails (for example the memory or PCB is full and cannot hold another process) then the fork call will generate a negative number.

A and B)



FCFS

P1 arrive
P2 arrive
P3 arrive
P4 arrive
P5 arrive
P1
P2
P3
P4
P5
0
4
8
12
16
20
24
28
32

Process	Arrival	Completion	Turnaround
1	0ms	12 ms	12 ms
2	5 ms	20 ms	15 ms
3	8 ms	23 ms	15 ms
4	15 ms	29 ms	14 ms
5	20 ms	34 ms	14 ms

Mean turnaround

$\left(\frac{12+15+15+14+14}{5}\right)ms$

$= 14\ ms$

RR

P1 arrive
P2 arrive
P3 arrive
P4 arrive
P5 arrive
P1
P1
P2
P3
P4
P1
P2
P4
P5
P5
0
4
8
12
16
20
24
28
32

Process	time left
1	~~12~~ ~~8~~ ~~4~~ 0
2	~~8~~ ~~4~~ 0
3	~~3~~ 0
4	~~6~~ ~~2~~ 0
5	~~5~~ ~~1~~ 0

Process	Arrival	Completion	Turnaround
1	0ms	23 ms	23 ms
2	5 ms	27 ms	22 ms
3	8 ms	15 ms	7 ms
4	15 ms	29 ms	14 ms
5	20 ms	34 ms	14 ms

Mean turnaround

$\left(\frac{23+22+3+14+14}{5}\right)ms = 16\ ms$

SJF w/ Premtion

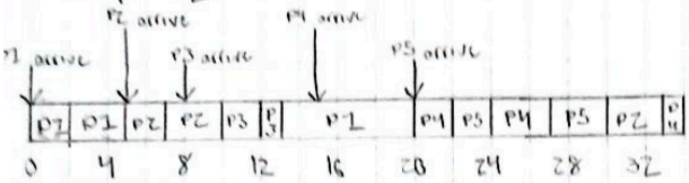
Process	time left
1	~~12~~ ~~11~~ 0
2	~~8~~ ~~5~~ ~~1~~ 0
3	~~3~~ 0
4	~~6~~ ~~1~~ 0
5	~~5~~ 0

Process	Arrival	Completion	Turnaround
1	0 ms	34 ms	34 ms
2	5 ms	27 ms	22 ms
3	8 ms	11 ms	3 ms
4	15 ms	26 ms	11 ms
5	20 ms	25 ms	5 ms

Mean turnaround

$\left(\frac{34+22+3+11+5}{6}\right)ms = 15ms$

Multiple queue



execution time remaining

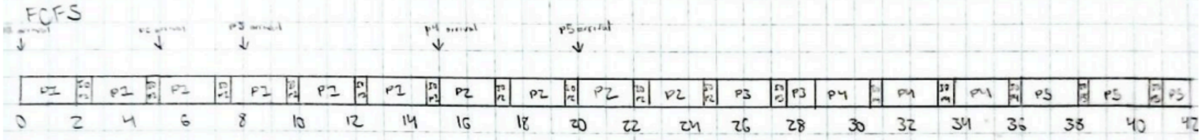
Process	Start Time	Q1 Time	Q2 Time	Q3 Time
1	12 ms	2 ms	3 ms	7 ms
2	8 ms	2 ms	3 ms	3 ms
3	3 ms	2 ms	1 ms	0 ms
4	6 ms	2 ms	3 ms	1 ms
5	5 ms	2 ms	3 ms	0 ms

Process	Arrival	Completion	Turnaround
1	0 ms	20 ms	20 ms
2	5 ms	33 ms	28 ms
3	8 ms	13 ms	5 ms
4	15 ms	34 ms	19 ms
5	20 ms	30 ms	10 ms

Mean Turnaround

$$\left(\frac{20+20+5+19+10}{5}\right) ms = \quad 16.4 \ ms$$

FCFS

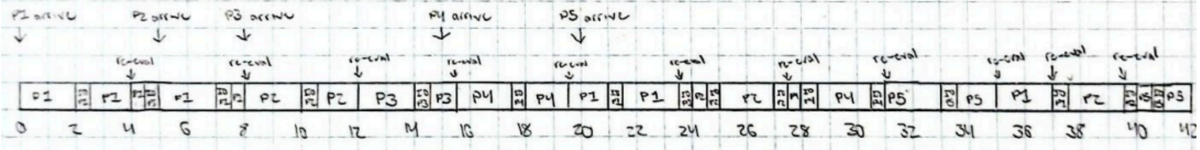


Process	Arrival	Completion	Turnaround
1	0 ms	14.5 ms	14.5 ms
2	5 ms	24.5 ms	19.5 ms
3	8 ms	28.5 ms	20.5 ms
4	15 ms	34.5 ms	19.5 ms
5	20 ms	42 ms	22 ms

Mean turnaround time

$$\left(\frac{14.5+19.5+20.5+19.5+22}{5}\right) ms = \quad 19.2 \ ms$$

RR



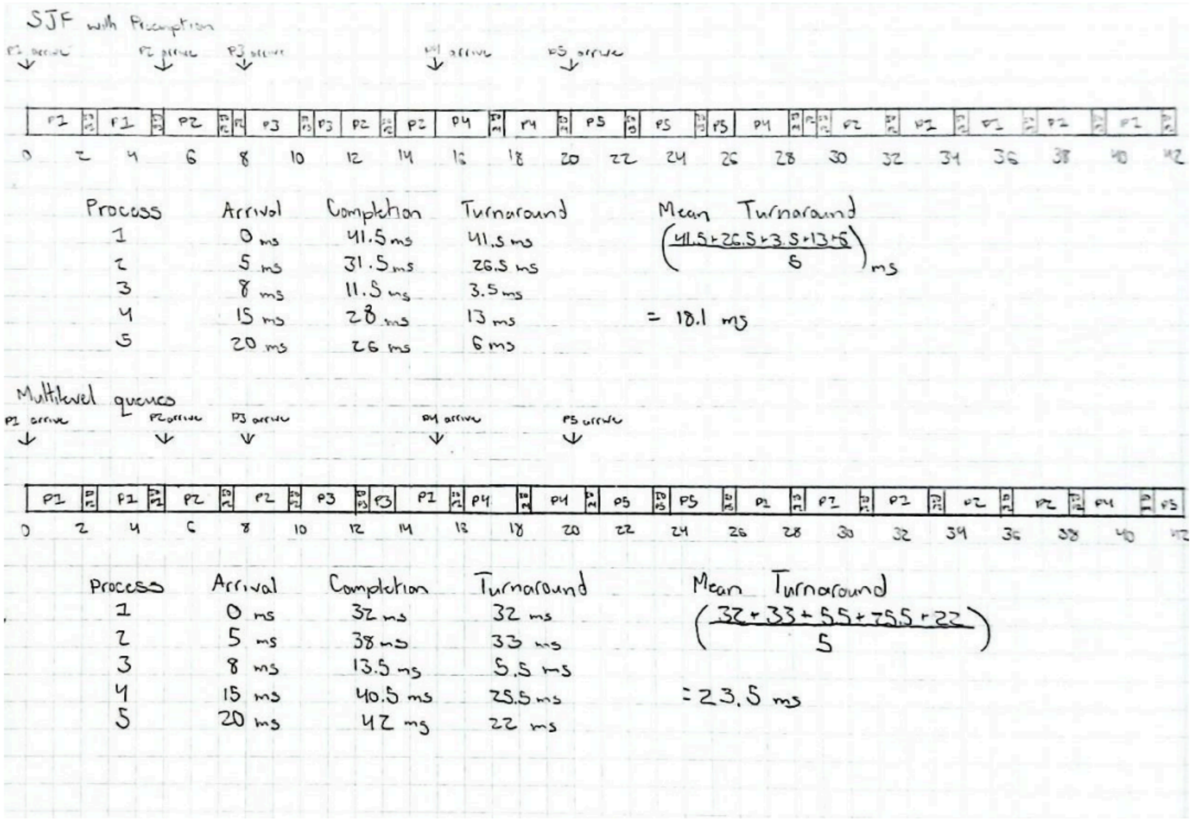
Process	Arrival	Completion	Turnaround
1	0 ms	37 ms	37 ms
2	5 ms	39.5 ms	34.5 ms
3	8 ms	15.5 ms	7.5 ms
4	15 ms	30.5 ms	15.5 ms
5	20 ms	42 ms	22 ms

Mean Turnaround

$$\left(\frac{37+34.5+7.5+15.5+22}{5}\right) ms$$

= 23.3 ms

Process	Time left
1	~~12~~ ~~9.5~~ ~~5.5~~ ~~2~~
2	~~8~~ ~~4.5~~ ~~2~~
3	~~3~~ 0
4	~~8~~ ~~2.5~~ 0
5	~~8~~ 1.5



SJF with Preemption

Process	Arrival	Completion	Turnaround
1	0 ms	41.5 ms	41.5 ms
2	5 ms	31.5 ms	26.5 ms
3	8 ms	11.5 ms	3.5 ms
4	15 ms	28 ms	13 ms
5	20 ms	26 ms	6 ms

Mean Turnaround

$\left(\frac{41.5+26.5+3.5+13+5}{5}\right)$ ms

= 18.1 ms

Multilevel queues

Process	Arrival	Completion	Turnaround
1	0 ms	32 ms	32 ms
2	5 ms	38 ms	33 ms
3	8 ms	13.5 ms	5.5 ms
4	15 ms	40.5 ms	25.5 ms
5	20 ms	42 ms	22 ms

Mean Turnaround

$\left(\frac{32+33+5.5+25.5+22}{5}\right)$

= 23.5 ms

C1 and C2:

First Fit:

Job #	Partition #
1	2
2	1
3	8
4	4
5	N/A

Partitions 3, 5, 6, 7 are free after allocation, these add up to 318 KB

Internal fragmentation: 200 + 3 + 5 + 130 = 338 KB

External fragmentation: 28 + 55 + 160 + 75 = 318 KB

Best fit:

Job #	Partition #
1	6
2	1
3	8
4	7
5	4

Partitions 2, 3, 5 are free after allocation, these add up 373 KB

Internal Fragmentation: 20 + 3 + 5 + 10 + 5 = 43 KB

External Fragmentation: 340 + 28 + 55 = 423 KB

Worst fit:

Job #	Partition #
1	2
2	8
3	N/A
4	4
5	N/A

Partitions 1, 3, 5, 6, 7 are free after allocation, these add up 403 KB

Internal Fragmentation: 200 + 198 + 130 = 528 KB

External Fragmentation: 85 + 28 + 55 + 160 + 75 = 403 KB

C) part 2: First fit has the best memory utilization, however best fit has by far the lowest internal fragmentation. This comes at a cost of slightly higher external fragmentation, and it takes longer as it must search for the smallest hole. From this data we can conclude that best fit would be the superior option for small allocations as first fit would have very high internal fragmentation in such a scenario. For a mixed payload, first fit would be better as its internal fragmentation would be closer to best fit and it would have better external fragmentation. Worst fit is not ideal for either scenario and is overall worse in any situation than the two algorithms mentioned above.

Step-by-step allocation:

First Fit:

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

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

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

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

No space left for Job 5

Best fit:

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

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

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

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

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

Worst fit:

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

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

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

No space for Jobs 3 and 5.