SYSC 4001
Assignment 2
November 13, 2023

Ali Abdollahian - 101229396
Nathan MacDiarmid - 101098993

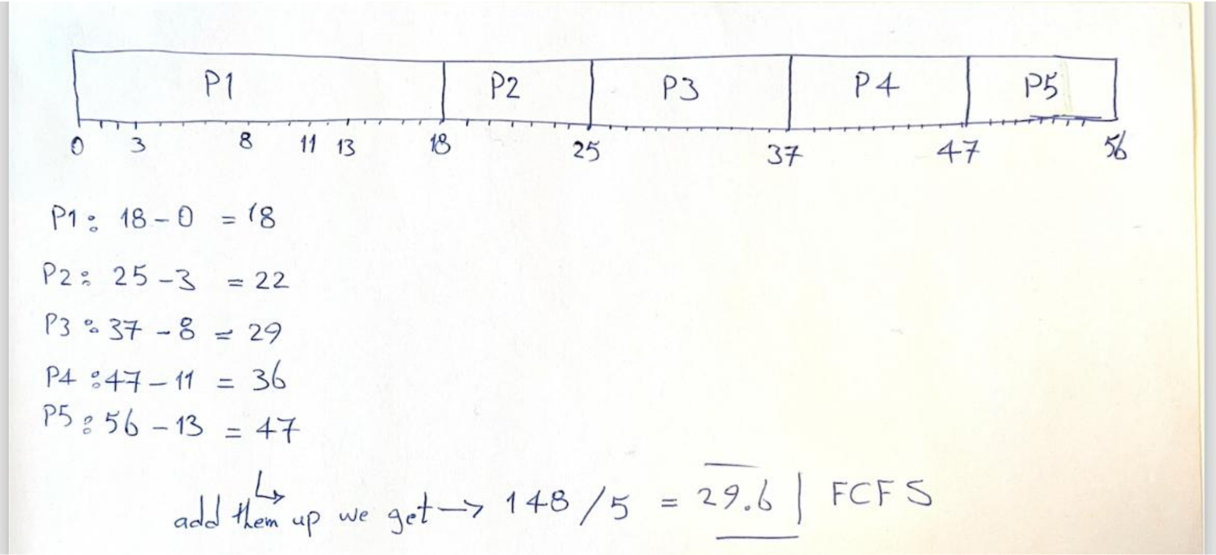
Part 1

a) The three possible events that can make a process abandon the use of the CPU are:

- The process finishes
- There is an interrupt (scheduler dispatch)
- There is an I/O event

State :	Why ?	How does the Kernel react ?
Running →Terminated	Process completes execution or user clicked termination button or is killed.	Kernel updates process control block then releases resources and also deallocates memory and lastly removes the process from the system.
Running →Ready	Time quantum exhausted or process preempted by a higher priority process.	Kernel initiates a context switch, saves the current process's state and moves it to the end of the ready queue, ready for future execution based on the scheduling algorithm in our case RR.
Running →Waiting	Process I/O requests	Kernel places the process in a waiting state then it updates its state in the PCB and schedules another process to execute while the process is executing the I/O event.

b)



P1	P2	P3	P4	P5

0 3 8 11 13 18 25 37 47 56

P1 : 18 − 0 = 18

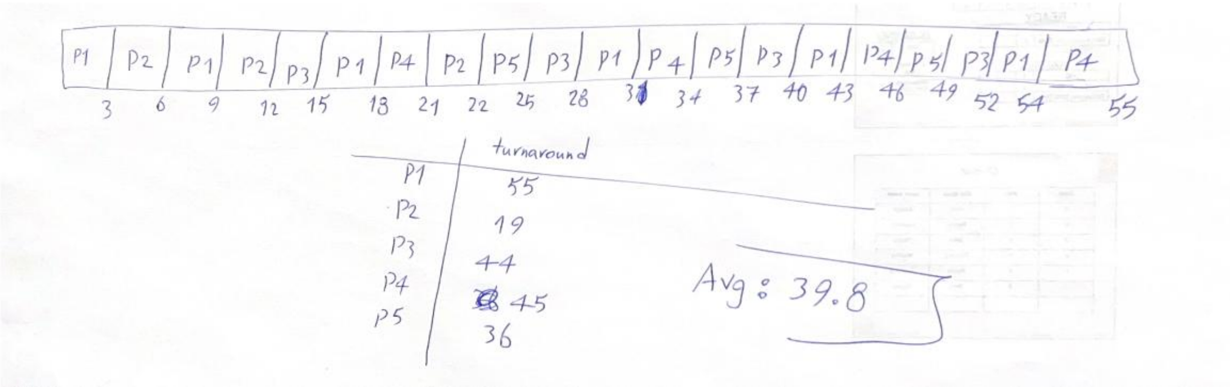
P2 : 25 − 3 = 22

P3 : 37 − 8 = 29

P4 : 47 − 11 = 36

P5 : 56 − 13 = 47

↳ add them up we get → 148 / 5 = 29.6 | FCFS

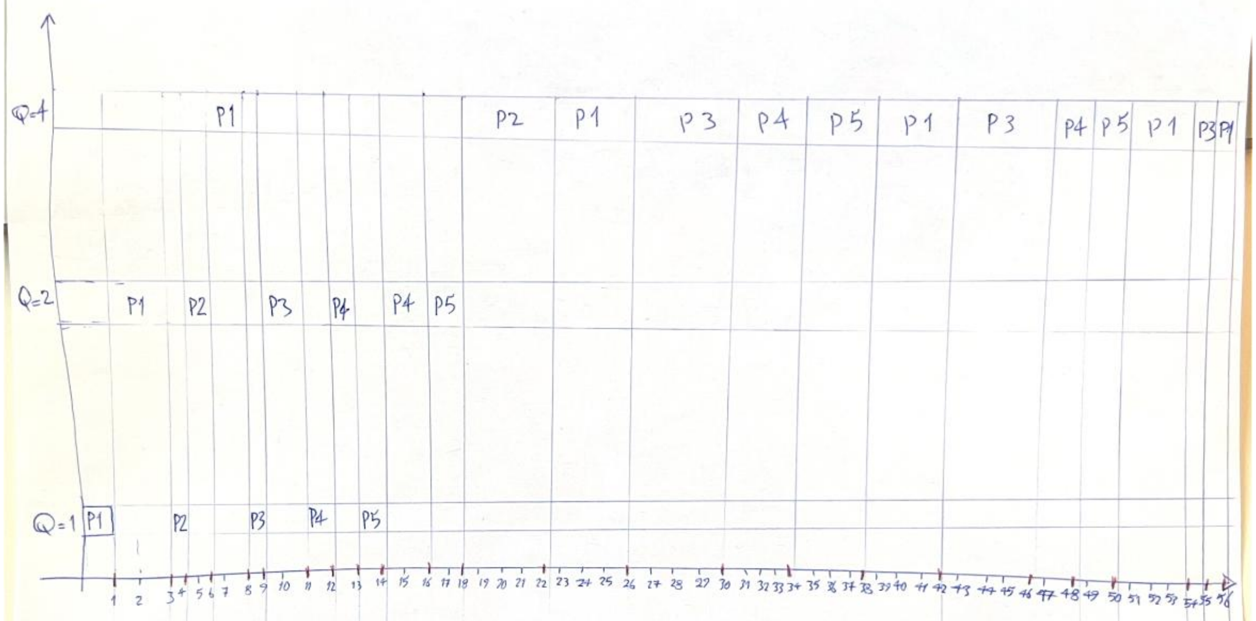


P1	P2	P1	P2	P3	P1	P4	P2	P5	P3	P1	P4	P5	P3	P1	P4	P5	P3	P1	P4

3 6 9 12 15 18 21 22 25 28 31 34 37 40 43 46 49 52 54 55

	turnaround
P1	55
P2	19
P3	44
P4	45
P5	36

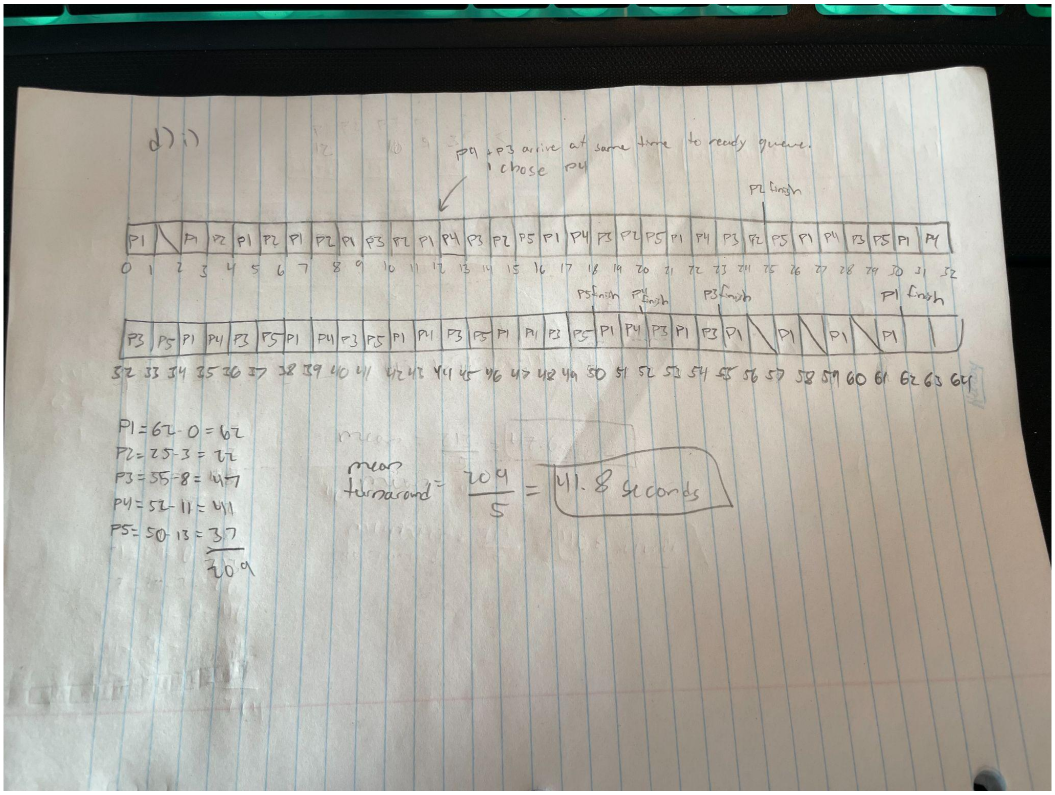
Avg : 39.8



Processes	Arrival	Completion	Turnaround time
P1	0	56	56
P2	3	22	19
P3	8	55	47
P4	11	48	37
P5	13	50	37

Average: $\boxed{39.2}$

d)



d) i)

P4 + P3 arrive at same time to ready queue. I chose P4

P2 finish

P1	\	P1	P2	P1	P2	P1	P2	P1	P3	P2	P1	P4	P3	P2	P5	P1	P4	P3	P2	P5	P1	P4	P3	P2	P5	P1	P4	P3	P5	P1	P4

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

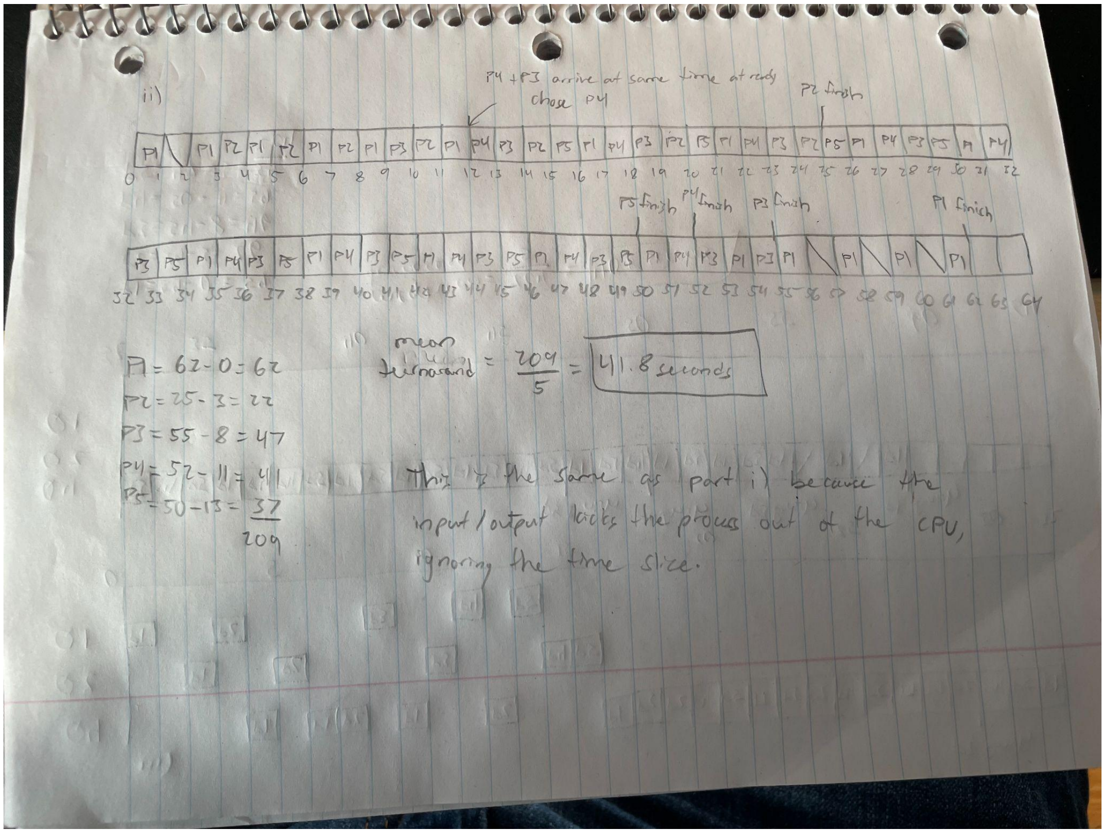
P5 finish P4 finish P3 finish P1 finish

P3	P5	P1	P4	P3	P5	P1	P4	P3	P5	P1	P4	P3	P5	P1	P4	P3	P5	P1	P4	P3	P1	P3	P1	\	P1	\	P1	\	P1		

32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64

P1 = 62 - 0 = 62
P2 = 25 - 3 = 22
P3 = 55 - 8 = 47
P4 = 52 - 11 = 41
P5 = 50 - 13 = 37
209

$$\text{mean turnaround} = \frac{209}{5} = \boxed{41.8 \text{ seconds}}$$



ii)

P4 + P3 arrive at same time at ready chose P4

P2 finish

P1	\	P1	P2	P1	P2	P1	P2	P1	P3	P2	P1	P4	P3	P2	P5	P1	P4	P3	P2	P5	P1	P4	P3	P2	P5	P1	P4	P3	P5	P1	P4

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

P5 finish P4 finish P3 finish P1 finich

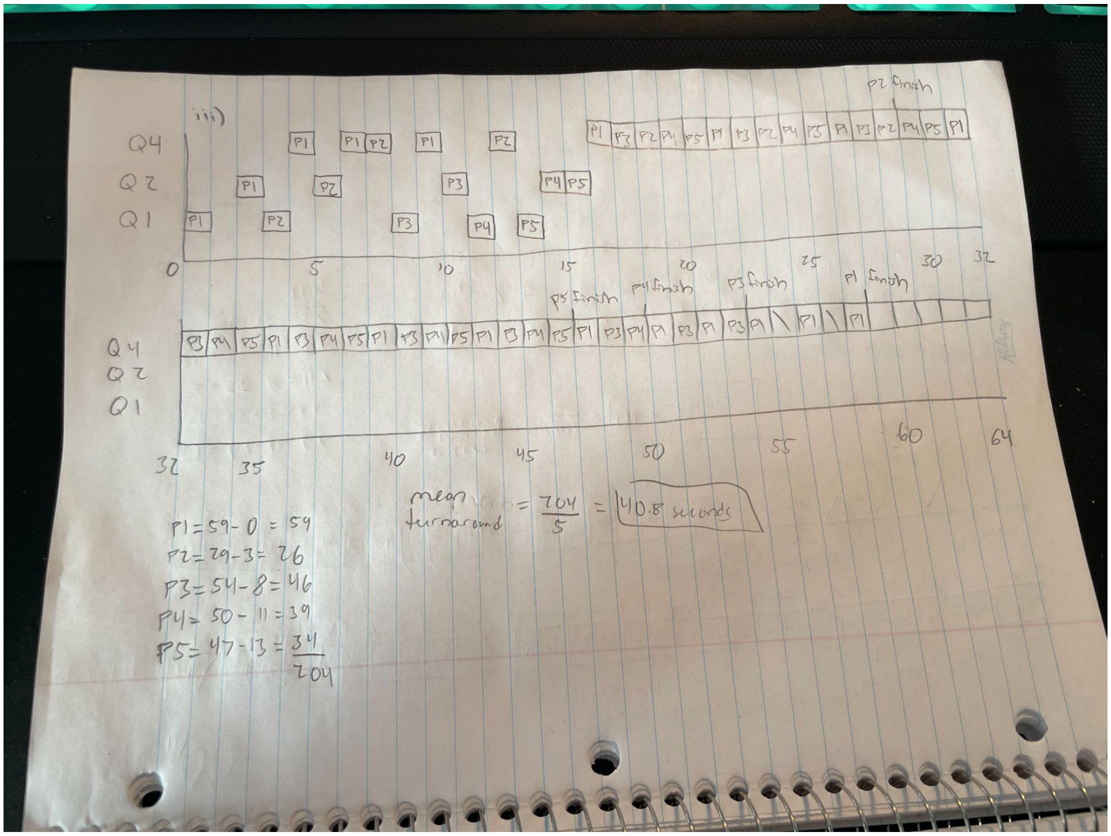
P3	P5	P1	P4	P3	P5	P1	P4	P3	P5	P1	P4	P3	P5	P1	P4	P3	P5	P1	P4	P3	P1	P3	P1	\	P1	\	P1	\	P1		

32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64

P1 = 62 - 0 = 62
P2 = 25 - 3 = 22
P3 = 55 - 8 = 47
P4 = 52 - 11 = 41
P5 = 50 - 13 = 37
209

$$\text{mean turnaround} = \frac{209}{5} = \boxed{41.8 \text{ seconds}}$$

This is the same as part i) because the input/output locks the process out of the CPU, ignoring the time slice.



P1 = 59 - 0 = 59
P2 = 29 - 3 = 26
P3 = 54 - 8 = 46
P4 = 50 - 11 = 39
P5 = 47 - 13 = 34
204

mean turnaround = $\frac{204}{5}$ = $\boxed{40.8 \text{ seconds}}$

e) **a) FCFS**

First Come First Serve is the least degree of discrimination toward longer processes as it is a queue that gives each process time to run in order of arrival. The discrimination comes when there are I/O events that must be triggered. This limits the amount of time a process will have the CPU, increasing the favorability of shorter processes as they will generally finish quicker with limited I/O events. It could also be the case that there is no discrimination toward longer processes in some cases because if there are no I/O events, all processes will run in order of arrival without exception.

b) RR

Round Robin is in between FCFS and multilevel feedback queues in degree of discrimination toward longer processes. This is because it relies on the FCFS nature of the queue of processes that need to be run, however, it implements a time slice which is a set amount of time that any one process can use the CPU. Once the time slice has been used, if the running process is not finished executing, it is moved back to the ready queue. In addition to the I/O events that favor shorter processes in FCFS, the time slice in RR further increases the discrimination toward longer processes because if they are not finished executing, or an I/O event doesn't happen, the process will be moved to the ready queue regardless.

c) Multilevel feedback queues

Multilevel feedback queues discriminate the most toward longer processes as it can have many different queues with many different priorities. All of these queues implement FCFS that already favors shorter processes when dealing with I/O events. Furthermore, each queue implements a time slice like RR, kicking a process out if it doesn't finish. However, the process is not moved back to the waiting queue like RR, instead it is moved from the highest priority queue (where it initially arrives) to the next highest priority queue. If there are no processes running in the highest priority queue, the next highest priority queue runs and this algorithm continues down the list of queues until a process is running, ensuring that a process is always running. Usually, the lower queues have longer time slices, allowing processes to run for longer. This is how multilevel feedback queues discriminate the most toward longer processes because if the process doesn't finish quickly, it will be moved to lower priority queues, giving shorter processes higher priority and allowing them to run first.

f)

a) First-Fit

- 122K is put in 205K partition
  leaving (102K, 83K (205K - 122K), 43K,180K, 70K, 125K, 91K and 150K)

- 105K is put in 180K partition
  leaving (102K, 83K (205K - 122K), 43K, 75K (180K - 105K), 70K, 125K, 91K and 150K)
- 203K Not Allocated / Must Wait

- 90K is put in 102K partition
  leaving (12K (102K - 90K), 83K (205K - 122K), 43K, 75K (180K - 105K), 70K, 125K, 91K and 150K)

b) Best-Fit

- 122K is put in 125K partition
  leaving (102K, 205K, 43K,180K, 70K, 3K (125K - 122K), 91K and 150K)

- 105K is put in 150K partition
  leaving (102K, 205K, 43K,180K, 70K, 3K (125K - 122K), 91K and 45K (150K - 105K))

- 203K is put in 205K partition
  leaving (102K, 2K (205K - 203K), 43K,180K, 70K, 3K (125K - 122K), 91K and 45K (150K - 105K))

- 90K is put in 91K partition leaving (102K, 2K (205K - 203K), 43K,180K, 70K, 3K (125K - 122K), 1K (91K - 90K) and 45K (150K - 105K))

c) Worst-Fit

- 122K is put in 205K partition
  leaving (102K, 83K (205K - 122K), 43K,180K, 70K, 125K, 91K and 150K)

- 105K is put in 180K partition
  leaving (102K, 83K (205K - 122K), 43K,75K (190K - 105K), 70K, 125K, 91K and 150K)

- 203K Not Allocated / Must Wait

- 90K is put in 150K partition
  leaving (102K, 83K (205K - 122K), 43K, 75K (190K - 105K), 70K, 125K, 91K and 60K (150K - 90K))

Part 2

iii)

Three Schedulers

The results obtained from these simulations are from a broad variety of tests given to our different schedulers in order to operate the functions of an operating system. The operating system for our simulations contains five different queues, each serving a different purpose. The new queue is used as a placeholder for all the processes that are loaded into the operating system to be run. They are all in a constant state of waiting until the cpu clock has reached the specified arrival time of a process contained in the new queue; if this happens, the process is moved from the new queue to the ready queue where it waits to be run on the processor. If the processor is free, a process is moved from the ready queue to the running queue where the executes its task until either an I/O event, or the process terminates. If an I/O event happens, the process is moved from the ready queue to the waiting queue where it has to execute its mandatory wait period performing some I/O function. Once the I/O function completes, it is then put back into the ready queue, waiting to be run on the processor again. If the process finishes running while in the processor, it will then be moved to the terminated queue, indicating that this process has finished and no longer requires access to the processor.

The three schedulers that were implemented to be compared are a First Come First Serve (FCFS) algorithm, Priority algorithm, and Round Robin (RR) algorithm. Each has different ways of handling how processes are loaded into the processor and how much time they are given to run once they are there. All three of them implement the queues depicted above and are all based on the FCFS algorithm regarding their queues.

The FCFS algorithm takes the first process of the ready queue, and places it in the processor to be executed. If there are no I/O events, that given process will run until completion. Otherwise, it will move to the waiting queue once the I/O event occurs and execute the I/O function and then it's put back into the end of the ready queue. While this is occurring, the FCFS algorithm has already moved the next process from the ready queue into the processor to be executed. This loop continues until all processes have been executed and moved to the terminated queue.

The Priority algorithm takes the process with the highest priority process from the ready queue and moves it to the processor to be executed. If there are no other processes in the ready queue, it takes the only one. The move from running to waiting and waiting to ready remains the same as the FCFS algorithm. In this algorithm, one is considered a high priority while five is considered a low priority. This loop also continues until all processes have been executed and moved to the terminated queue.

The RR queue takes the first process from the ready queue similar to the FCFS algorithm and places it in the processor to be executed. It is then assigned a time slice, which is the maximum amount of time it can remain in the processor, regardless of I/O events. If an I/O event were to be scheduled to happen every five seconds, and there is a time slice of four seconds assigned to the process that is running in the processor, the process will be moved to the waiting queue after four seconds, regardless if the I/O event would take place within the next second. The move from running to waiting and waiting to ready remains the same as the FCFS algorithm. The time slice for this algorithm was one second. The system loop also continues until all processes have been executed and moved to the terminated queue.

The results from these simulations measured the average turnaround time, average wait time, and throughput of all three algorithms given the same input. As depicted in Appendix 1, the results vary through the different algorithms, but there are common similarities and averages between them. For example, any time that the I/O frequency is the same as the time slice, FCFS and RR will be relatively similar if not identical. This is because the RR algorithm is based on the FCFS algorithm only changing the max amount of time a process can remain in the processor. Additionally, the total time it took to execute all the processes in the program stayed relatively similar between all three algorithms, with a minor degree of change of roughly 10-20 seconds. This also kept the throughput of all the algorithms relatively similar, changing just slightly as the total amount of time to execute all the processes changed. However, the average turnaround time verified greatly between algorithms with RR being the most inefficient for turnaround time. Unless the I/O frequency was the same as the time slice, RR took much longer than the FCFS and Priority algorithms.

The results depicted a clear divide in performance regarding the three metrics of average turnaround time, average wait time, and throughput. In regards to average turnaround time and average wait time, RR was the slowest even if the total time of executing was the same as FCFS. Priority was the middle performance wise while FCFS was the superior algorithm for average turnaround time and average wait time. The average turnaround time and average wait time were also an accurate depiction of throughput of the algorithm, with RR again being the least efficient, outputting less processes per time than both Priority and FCFS. Priority was again in the middle and FCFS being the superior algorithm outputting the most processes per time than both algorithms.

This simulation proved interesting regarding the efficiency and performance of each of the scheduling algorithms taught to us in class. It confirms that FCFS is the most efficient algorithm out of the three algorithms whereas RR is the worst in terms of output and turnaround time. However, it also shed light on how RR's wait times for each individual process were relatively similar, with there being multiple cases of wait time being the same for different processes. It also shed light that Priority will almost always finish in order of priority.

In conclusion, these findings were found to be very interesting, seeing first hand what is being taught in the classroom being proven by our own simulation. It also shows that each algorithm serves a purpose and can be used in various types of scenarios in order to accomplish a task, giving operating system developers various tools to use to solve said task.

iv)

Memory Management :

In this section, we implemented a First-Come-First-Serve (FCFS) scheduling algorithm with varying process parameters and memory requirements. The memory is divided into partitions of different sizes, and processes are allocated based on their arrival times, CPU time, I/O frequency, I/O duration, and memory needs.

Memory Partition Configuration 1

In the initial simulation, memory partitions were set to 500 MB, 250 MB, 150 MB, and 100 MB. The FCFS scheduler was used. The results are as follows:

- Memory usage varied across processes, with larger memory-demanding test case processes utilizing more space and this configuration of memory partition had a better outcome.
- Partitions were utilized based on the memory needs of each process.
- The total free memory was very varied since processes were allocated and deallocated from memory. It was affected by the sizes of the processes and the available memory partitions.
- Usable memory, excluding internal fragmentation, was the sum of the available memory in all partitions. This metric reflects the actual memory space usable by processes.

Memory Partition Configuration 2

In the second simulation, memory partitions were adjusted to 300 MB, 300 MB, 350 MB, and 50 MB. The same FCFS scheduler was used. The results are as follows:

- Similar to the first configuration, memory usage varied across processes, with larger processes consuming more space but this time test cases with higher memory had a higher waiting time because the partitions were smaller.
- The partition sizes influenced the distribution of processes. Smaller partitions led to potential memory allocation issues for larger processes.
- The total free memory fluctuated, influenced by the sizes of processes and the availability of partitions. Smaller partitions led to a more fragmented memory space.
- Usable memory, excluding internal fragmentation, reflected the actual memory space available for processes. Smaller partitions contributed to increased internal fragmentation.

Analysis

Impact of Partition Size on Memory Usage

- Smaller partitions led to increased internal fragmentation, reducing memory allocation efficiency. Larger partitions allowed for more flexible allocation of larger processes.
- Smaller partitions resulted in longer waiting times for processes with higher memory demands in the second 5 test cases. The system struggled to allocate contiguous memory space for larger processes.
- Larger partitions facilitated the efficient execution of processes, improving system throughput. Smaller partitions led to more frequent memory allocation delays and increased contention.

Conclusion

The simulation results highlight the critical role of memory partition sizes in determining system performance. The choice of partition sizes directly influences memory allocation efficiency, waiting times, and overall system throughput. Smaller partitions may lead to increased internal fragmentation and longer waiting times, impacting system performance. Larger partitions, while reducing fragmentation, require careful management to ensure optimal memory utilization. The analysis highlights the importance of aligning memory partition configurations with the specific characteristics of the workload to achieve optimal system performance.

Appendices

Appendix 1

Values (all values are in seconds)
Priority is 1 = high and 5 = low
Round Robin time slice is 1 second

First Come First Serve

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	5	1	1	9	4
1002	5	10	1	1	26	16
1003	10	15	1	1	53	38
1004	15	20	1	1	71	51
1005	20	25	1	1	80	55
1006	25	30	1	1	88	58

Total time = 113
Throughput = 6/113
Average turnaround = 54.5
Average wait time = 37

Priority

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	5	1	1	9	4
1002	5	10	1	1	19	9
1003	10	15	1	1	29	14
1004	15	20	1	1	49	29
1005	20	25	1	1	69	44
1006	25	30	1	1	99	69

Total time = 124
Throughput = 6/124

Average turnaround = 191.5
Average wait time = 28.17

Round Robin

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	5	1	1	9	4
1002	5	10	1	1	26	16
1003	10	15	1	1	53	38
1004	15	20	1	1	71	51
1005	20	25	1	1	80	55
1006	25	30	1	1	88	58

Total time = 113
Throughput = 6/113
Average turnaround = 54.5
Average wait time = 37

First Come First Serve

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	5	2	1	8	3
1002	5	10	2	1	19	9
1003	10	15	2	1	51	36
1004	15	20	2	1	70	50
1005	20	25	2	1	80	55
1006	25	30	2	1	83	53

Total time = 108
Throughput = 6/108
Average turnaround = 51.83
Average wait time = 34.33

Priority

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	5	2	1	8	3
1002	5	10	2	1	17	7
1003	10	15	2	1	29	14
1004	15	20	2	1	46	26
1005	20	25	2	1	70	45
1006	25	30	2	1	88	58

Total time = 113
Throughput = 6/113
Average turnaround = 43
Average wait time = 25.5

Round Robin

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	5	2	1	9	4
1002	5	10	2	1	26	16
1003	10	15	2	1	53	38
1004	15	20	2	1	71	51
1005	20	25	2	1	80	55
1006	25	30	2	1	88	58

Total time = 113
Throughput = 6/113
Average turnaround = 54.5
Average wait time = 37

First Come First Serve

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	15	20	2	3	72	52
1002	5	15	3	2	41	26
1003	10	5	4	3	15	10
1004	15	2	5	2	9	7
1005	20	6	6	3	11	5
1006	25	14	7	2	30	16

Total time = 87

Throughput = 6/87

Average turnaround = 29.67

Average wait time = 19.33

Priority

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	15	20	2	3	60	40
1002	5	15	3	2	27	12
1003	10	5	4	3	25	20
1004	15	2	5	2	22	20
1005	20	6	6	3	25	19
1006	25	14	7	2	38	24

Total time = 75

Throughput = 6/75

Average turnaround = 32.83

Average wait time = 22.5

Round Robin

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	15	20	2	3	80	60
1002	5	15	3	2	56	41
1003	10	5	4	3	19	14
1004	15	2	5	2	7	5
1005	20	6	6	3	25	19
1006	25	14	7	2	55	41

Total time = 95
Throughput = 6/95
Average turnaround = 40.33
Average wait time = 30

First Come First Serve

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	2	2	2	8	2	0
1002	7	7	5	5	22	15
1003	6	6	3	3	19	13
1004	3	3	3	6	4	1
1005	5	5	5	7	7	2
1006	8	8	2	2	27	19

Total time = 35
Throughput = 6/35
Average turnaround = 13.5
Average wait time = 8.33

Priority

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	2	2	2	8	2	0
1002	7	7	5	5	15	8
1003	6	6	3	3	19	13
1004	3	3	3	6	4	1
1005	5	5	5	7	15	10
1006	8	8	2	2	31	23

Total time = 39
Throughput = 6/39
Average turnaround = 14.33
Average wait time = 9.17

Round Robin

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	2	2	2	8	11	9
1002	7	7	5	5	40	33
1003	6	6	3	3	24	18
1004	3	3	3	6	16	13
1005	5	5	5	7	35	30
1006	8	8	2	2	29	21

Total time = 47
Throughput = 6/47
Average turnaround = 25.83
Average wait time = 20.67

First Come First Serve

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	55	1	1	296	241
1002	5	60	1	1	331	271
1003	10	65	1	1	356	291
1004	15	70	1	1	372	302
1005	20	75	1	1	380	305
1006	25	80	1	1	388	308

Total time = 413
Throughput = 6/413
Average turnaround = 353.83
Average wait time = 286.33

Priority

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	55	1	1	109	54
1002	5	60	1	1	119	59
1003	10	65	1	1	229	164
1004	15	70	1	1	249	179
1005	20	75	1	1	369	294
1006	25	80	1	1	399	319

Total time = 424
Throughput = 6/424
Average turnaround = 245.67
Average wait time = 178.17

Round Robin

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	55	1	1	296	241
1002	5	60	1	1	331	271
1003	10	65	1	1	356	291
1004	15	70	1	1	372	302
1005	20	75	1	1	380	305
1006	25	80	1	1	388	308

Total time = 413

Throughput = 6/413

Average turnaround = 353.83

Average wait time = 286.33

First Come First Serve

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	5	1	2	13	8
1002	7	10	1	3	42	32
1003	12	15	1	4	66	51
1004	4	20	1	5	145	125
1005	13	25	1	6	191	166
1006	18	30	1	7	233	203

Total time = 251

Throughput = 6/251

Average turnaround = 115

Average wait time = 97.5

Priority

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	5	1	2	13	8
1002	7	10	1	3	37	27
1003	12	15	1	4	58	43
1004	4	20	1	5	115	95
1005	13	25	1	6	181	156
1006	18	30	1	7	242	212

Total time = 260
Throughput = 6/260
Average turnaround = 107.67
Average wait time = 90.17

Round Robin

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	5	1	2	13	8
1002	7	10	1	3	42	32
1003	12	15	1	4	66	51
1004	4	20	1	5	145	125
1005	13	25	1	6	191	166
1006	18	30	1	7	233	203

Total time = 251
Throughput = 6/251
Average turnaround = 115
Average wait time = 97.5

First Come First Serve

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	100	1	15	1585	1485
1002	5	200	2	30	3170	2970

Total time = 3175

Throughput = 2/3175

Average turnaround = 951

Average wait time = 2227.5

Priority

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	100	1	15	1585	1485
1002	5	200	2	30	3170	2970

Total time = 3175

Throughput = 2/3175

Average turnaround = 951

Average wait time = 2227.5

Round Robin

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	100	1	15	1589	1489
1002	5	200	2	30	6170	5970

Total time = 6175

Throughput = 2/6175

Average turnaround = 3879.5

Average wait time = 3729.5

First Come First Serve

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	32	5	1	209	177
1002	5	33	10	1	190	157
1003	10	34	15	1	182	148
1004	15	35	20	1	140	105
1005	20	36	25	1	156	120
1006	25	37	30	1	163	126

Total time = 209

Throughput = 6/209

Average turnaround = 173.33

Average wait time = 138.83

Priority

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	32	5	1	95	63
1002	5	33	10	1	48	15
1003	10	34	15	1	89	55
1004	15	35	20	1	144	109
1005	20	36	25	1	150	114
1006	25	37	30	1	183	146

Total time = 208

Throughput = 6/208

Average turnaround = 118.17

Average wait time = 83.67

Round Robin

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	32	5	1	158	126
1002	5	33	10	1	173	140
1003	10	34	15	1	182	148
1004	15	35	20	1	186	151
1005	20	36	25	1	186	150
1006	25	37	30	1	186	149

Total time = 211

Throughput = 6/211

Average turnaround = 178.5

Average wait time = 144

First Come First Serve

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	10	4	5	38	28
1002	2	10	4	10	42	32
1003	4	10	4	15	49	39
1004	6	10	4	20	58	48
1005	8	10	4	25	68	58
1006	10	10	4	30	84	74

Total time = 94

Throughput = 6/94

Average turnaround = 56.5

Average wait time = 46.5

Priority

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	10	4	5	26	16
1002	2	10	4	10	34	24
1003	4	10	4	15	47	37
1004	6	10	4	20	60	50
1005	8	10	4	25	78	68
1006	10	10	4	30	96	86

Total time = 106
Throughput = 6/106
Average turnaround = 56.83
Average wait time = 46.83

Round Robin

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	10	4	5	57	47
1002	2	10	4	10	103	93
1003	4	10	4	15	145	135
1004	6	10	4	20	193	183
1005	8	10	4	25	235	225
1006	10	10	4	30	280	270

Total time = 290
Throughput = 6/290
Average turnaround = 168.83
Average wait time = 158.83

First Come First Serve

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	10	5	15	33	23
1002	2	10	2	6	56	46
1003	4	10	3	8	52	42
1004	6	10	2	15	83	73
1005	8	10	4	20	58	48
1006	10	10	7	2	26	16

Total time = 89

Throughput = 6/89

Average turnaround = 51.33

Average wait time = 41.33

Priority

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	10	5	15	26	16
1002	2	10	2	6	49	39
1003	4	10	3	8	39	29
1004	6	10	2	15	77	67
1005	8	10	4	20	61	51
1006	10	10	7	2	44	34

Total time = 83

Throughput = 6/83

Average turnaround = 49.33

Average wait time = 39.33

Round Robin

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Turnaround	Wait time
1001	0	10	5	15	146	136
1002	2	10	2	6	66	56
1003	4	10	3	8	86	76
1004	6	10	2	15	145	135
1005	8	10	4	20	191	181
1006	10	10	7	2	35	25

Total time = 199

Throughput = 6/199

Average turnaround = 111.5

Average wait time = 101.5

Appendix 2

Memory Management

Part 1 :

Memory partitions :

```
#define MEMORY_PARTITION_1 500
#define MEMORY_PARTITION_2 250
#define MEMORY_PARTITION_3 150
#define MEMORY_PARTITION_4 100
```

FCFS

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Memory Needed
1001	0	5	1	1	100
1002	5	10	1	1	10
1003	10	15	1	1	250
1004	15	20	1	1	90
1005	20	25	1	1	1000
1006	25	30	1	1	300

```
Process 1001 - Turnaround Time: 9 units, Wait Time: 4 units
Process 1002 - Turnaround Time: 24 units, Wait Time: 14 units
Process 1003 - Turnaround Time: 34 units, Wait Time: 19 units
Process 1004 - Turnaround Time: 44 units, Wait Time: 24 units
Process 0 - Turnaround Time: 0 units, Wait Time: 0 units
Process 1006 - Turnaround Time: 79 units, Wait Time: 49 units
```

FCFS

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Memory Needed
1001	0	5	2	1	100
1002	5	10	2	1	1000
1003	10	15	2	1	250
1004	15	20	2	1	350
1005	20	25	2	1	450
1006	25	30	2	1	50

Process 1001 - Turnaround Time: 7 units, Wait Time: 2 units
Process 0 - Turnaround Time: 0 units, Wait Time: 0 units
Process 1003 - Turnaround Time: 23 units, Wait Time: 8 units
Process 1004 - Turnaround Time: 58 units, Wait Time: 38 units
Process 1005 - Turnaround Time: 94 units, Wait Time: 69 units
Process 1006 - Turnaround Time: 58 units, Wait Time: 28 units

FCFS

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Memory Needed
1001	15	20	2	3	2000
1002	5	15	3	2	300
1003	10	5	4	3	200
1004	15	2	5	2	40
1005	20	6	6	3	450
1006	25	14	7	2	5000

Process 0 - Turnaround Time: 0 units, Wait Time: 0 units
Process 1002 - Turnaround Time: 25 units, Wait Time: 10 units
Process 1003 - Turnaround Time: 12 units, Wait Time: 7 units
Process 1004 - Turnaround Time: 6 units, Wait Time: 4 units
Process 1005 - Turnaround Time: 16 units, Wait Time: 10 units

Process 0 - Turnaround Time: 0 units, Wait Time: 0 units

FCFS

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Memory Needed
1001	1	4	2	8	40
1002	7	7	5	5	390
1003	6	6	3	3	300
1004	3	3	3	6	100
1005	5	5	5	7	200
1006	8	8	2	2	50

Process 1001 - Turnaround Time: 15 units, Wait Time: 11 units
Process 1002 - Turnaround Time: 14 units, Wait Time: 7 units
Process 1003 - Turnaround Time: 15 units, Wait Time: 9 units
Process 1004 - Turnaround Time: 6 units, Wait Time: 3 units
Process 1005 - Turnaround Time: 18 units, Wait Time: 13 units
Process 1006 - Turnaround Time: 26 units, Wait Time: 18 units

FCFS

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Memory Needed
1001	0	55	1	1	400
1002	5	60	1	1	400
1003	10	65	1	1	400
1004	15	70	1	1	400
1005	20	75	1	1	400
1006	25	80	1	1	400

Process 1001 - Turnaround Time: 109 units, Wait Time: 54 units
Process 1002 - Turnaround Time: 223 units, Wait Time: 163 units
Process 1003 - Turnaround Time: 794 units, Wait Time: 729 units
Process 1004 - Turnaround Time: 660 units, Wait Time: 590 units
Process 1005 - Turnaround Time: 516 units, Wait Time: 441 units

Process 1006 - Turnaround Time: 362 units, Wait Time: 282 units

Part 2:

Memory partitions :

```
#define MEMORY_PARTITION_1 300
#define MEMORY_PARTITION_2 300
#define MEMORY_PARTITION_3 350
#define MEMORY_PARTITION_4 50
```

FCFS

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Memory Needed
1001	0	55	1	1	300
1002	5	60	1	1	300
1003	10	65	1	1	300
1004	15	70	1	1	300
1005	20	75	1	1	300
1006	25	80	1	1	300

Process 1001 - Turnaround Time: 159 units, Wait Time: 104 units
Process 1002 - Turnaround Time: 176 units, Wait Time: 116 units
Process 1003 - Turnaround Time: 193 units, Wait Time: 128 units
Process 1004 - Turnaround Time: 354 units, Wait Time: 284 units
Process 1005 - Turnaround Time: 389 units, Wait Time: 314 units
Process 1006 - Turnaround Time: 379 units, Wait Time: 299 units

FCFS

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Memory Needed
1001	0	100	1	15	350
1002	5	200	2	30	350

```
Process 1001 - Turnaround Time: 1585 units, Wait Time: 1485 units
Process 1002 - Turnaround Time: 4750 units, Wait Time: 4550 units
```

FCFS

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Memory Needed
1001	0	32	5	1	300
1002	5	33	10	1	150
1003	10	34	15	1	400
1004	15	35	20	1	20
1005	20	36	25	1	300
1006	25	37	30	1	350

```
Process 1001 - Turnaround Time: 166 units, Wait Time: 134 units
Process 1002 - Turnaround Time: 108 units, Wait Time: 75 units
Process 0 - Turnaround Time: 0 units, Wait Time: 0 units
Process 1004 - Turnaround Time: 55 units, Wait Time: 20 units
Process 1005 - Turnaround Time: 109 units, Wait Time: 73 units
Process 1006 - Turnaround Time: 148 units, Wait Time: 111 units
```

FCFS

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Memory Needed
1001	0	10	4	5	140
1002	4	10	4	10	100
1003	4	10	4	15	20
1004	6	10	4	20	40
1005	8	10	4	25	50
1006	10	10	4	30	40

```
Process 1001 - Turnaround Time: 51 units, Wait Time: 41 units
Process 1002 - Turnaround Time: 49 units, Wait Time: 39 units
Process 1003 - Turnaround Time: 51 units, Wait Time: 41 units
Process 1004 - Turnaround Time: 65 units, Wait Time: 55 units
Process 1005 - Turnaround Time: 113 units, Wait Time: 103 units
Process 1006 - Turnaround Time: 120 units, Wait Time: 110 units
```

FCFS

PID	Arrival Time	Total CPU Time	I/O Freq	I/O Duration	Memory Needed
1001	0	10	0	0	140
1002	2	10	0	0	100
1003	3	13	0	0	200
1004	6	9	0	0	400
1005	8	10	0	0	29
1006	12	10	0	0	29

```
Process 1001 - Turnaround Time: 10 units, Wait Time: 0 units
Process 1002 - Turnaround Time: 18 units, Wait Time: 8 units
Process 1003 - Turnaround Time: 30 units, Wait Time: 17 units
Process 0 - Turnaround Time: 0 units, Wait Time: 0 units
Process 1005 - Turnaround Time: 35 units, Wait Time: 25 units
Process 1006 - Turnaround Time: 41 units, Wait Time: 31 units
```