



**UNIVERSITI  
MALAYA**

**LAB 5 : DEADLOCK MANAGEMENT  
TECHNIQUE**

**GROUP: 10**

**GROUP NAME: PKP**

<b>Name</b>	<b>Matric Number</b>
Afiq Ammar bin Ali Azhar	17069817/2
Adam Irfaan Bin Zambre	17139024/2
Amierul Haadif bin Azhari	17205070/1
Afiq Hakimi Bin Shairul Baki	17175291/2
Wan Arief Hakemi bin Darmafikri	17193642/2

## **LAB 5 : DEADLOCK MANAGEMENT TECHNIQUE**

Write a program to simulate disk scheduling algorithms

a) FCFS

### **FCFS Disk Scheduling Algorithms Concept**

First Come First Serve (FCFS) is the simplest disk scheduling algorithm. As the name suggests, this algorithm entertains requests in the order they arrive in the disk queue. The algorithm looks very fair and there is no starvation (all requests are serviced sequentially) but generally, it does not provide the fastest service.

## FCFS Solution

```
Lab 5 > FCFS.py > ...
1  def FCFS(request, currentPosition):
2      seekTime = 0 # initialize seek time
3
4      for i in range(0, len(request)):
5          # head is set to currentPosition
6          time = abs(request[i] - currentPosition)
7          # increment seek time by with the absolute distance of the track from the head.
8          seekTime = seekTime + time
9          print("Current Position:", currentPosition, "Time taken:", time, "Request:", request[i], "Seek Time:", seekTime)
10         currentPosition = request[i] # set head to current track
11
12     print("Total Seek Time: ", seekTime)
13     average = seekTime/len(request) # total seek time divided by track array length
14     print("Average Seek Time:", average)
15
16 request = [176, 79, 34, 60, 92, 11, 41, 114] # initialize tracks
17 currentPosition = 48 # initialize current position (position of the disk head)
18 FCFS(request, currentPosition)
```

### Explanation of the code above:

#### Line 1-11

Initialize seek time equal to zero. For loop from 0 to length of the request array to calculate total seek time of the disk scheduling.

Set the head to the current position(position of the disk head).

Set the time equal to the absolute value of the distance of the current track and head

Increment seek time by the absolute distance of the track from the head.

Set the head to the current track.

The average is calculated by dividing the seek time and length of the track array.

#### Line 13-15

Initialize an array of tracks and position of the disk head. Call the FCFS function to calculate seek time and average seek time.

## FCFS Disk Scheduling

```
PS C:\Users\afiq\Desktop\Bachelor\SEM 4\Operating System WIA2004\Operating System Lab> & C:/Users/afiq/AppData/Local/Programs/Python/Python38/python.exe "c:/Users/afiq/Desktop/Bachelor/SEM 4/Operating System WIA2004/Operating System Lab/Lab 5/FCFS.py"
Current Position: 48 Time taken: 128 Request: 176 Seek Time: 128
Current Position: 176 Time taken: 97 Request: 79 Seek Time: 225
Current Position: 79 Time taken: 45 Request: 34 Seek Time: 270
Current Position: 34 Time taken: 26 Request: 60 Seek Time: 296
Current Position: 60 Time taken: 32 Request: 92 Seek Time: 328
Current Position: 92 Time taken: 81 Request: 11 Seek Time: 409
Current Position: 11 Time taken: 30 Request: 41 Seek Time: 439
Current Position: 41 Time taken: 73 Request: 114 Seek Time: 512
Total Seek Time: 512
Average Seek Time: 64.0
PS C:\Users\afiq\Desktop\Bachelor\SEM 4\Operating System WIA2004\Operating System Lab>
```

Based on the output above, we can see that the seek time is incremented by the total distance from the track and head (Time taken - current position). Also, we can see that the head is set to current track after the seek time is being incremented. The average also is being calculated and printed.

## List of Contributions

Name	Contributions
Afiq Ammar bin Ali Azhar	<ul style="list-style-type: none"><li>- Finalised First Come First Serve algorithm.</li><li>- Contributor for First Come First Serve algorithm.</li><li>- Presentation on First Come First Serve algorithm.</li></ul>
Adam Irfaan Bin Zambre	<ul style="list-style-type: none"><li>- Finalised First Come First Serve algorithm.</li><li>- Presentation on First Come First Serve algorithm.</li></ul>
Amierul Haadif bin Azhari	<ul style="list-style-type: none"><li>- Finalised First Come First Serve algorithm.</li></ul>
Afiq Hakimi Bin Shairul Baki	<ul style="list-style-type: none"><li>- Finalised First Come First Serve algorithm.</li></ul>
Wan Arief Hakemi bin Darmafikri	<ul style="list-style-type: none"><li>- Finalised First Come First Serve algorithm.</li></ul>