**DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING**

**COLLEGE OF E&ME, NUST, RAWALPINDI**

# OS Project

Disk Scheduling Algorithms

| Submission to |
|---|
| Dr. Mehwish Naseer |

| Name | Registration # | Degree |
|---|---|---|
| Muhammad Sumair | 415339 | CE-44-A |
| Usama | 424024 | CE-44-A |
| Sher M. Behzad | 431984 | CE-44-A |

| Submission Date |
|---|
| 26 May 2024 |

## Introduction:

Disk scheduling algorithms play a crucial role in optimizing the performance of disk I/O operations by efficiently managing the movement of the disk arm. In this project, we implemented various disk scheduling algorithms and developed a graphical user interface (GUI) for users to visualize and compare these algorithms.

## Objectives:

The primary objectives of this project were as follows:

1. Implementation of popular disk scheduling algorithms:

   - First-Come, First-Served (FCFS)
   - Shortest Seek Time First (SSTF)
   - SCAN
   - Circular SCAN (C-SCAN)
   - LOOK
   - Circular LOOK (C-LOOK)

2. Creation of a user-friendly GUI for interactive visualization of disk scheduling algorithms.

3. Comparative analysis of the algorithms based on total head movement and execution time.

## Implementation

**Algorithms Implemented**

1. **First-Come, First-Served (FCFS):** Processes requests in the order they arrive.

2. **Shortest Seek Time First (SSTF):** Services the request closest to the current head position first.

3. **SCAN**: Moves the disk arm from one end to the other, serving requests along the way, then reverses direction.

4**. Circular SCAN (C-SCAN):** Like SCAN but only moves in one direction, jumping from the last position to the first after reaching the end.

5. **LOOK:** Like SCAN but does not travel the full extent of the disk unless necessary.

6. **Circular LOOK (C-LOOK):** Similar to LOOK but only moves in one direction, jumping from the last position to the first after reaching the end.

## Graphical User Interface (GUI)

```
import customtkinter as ctk
```

- Developed using **customtkinter** library for Python, providing a dark-themed, user-friendly interface.
- Allows users to select an algorithm, input request sequence, specify the current position of the disk arm, and set the unit time.
- Provides options to visualize the movement of the disk arm and generate a bar chart comparing total head movement for each algorithm.

## Visualization

```
import turtle
```

- Utilized Turtle graphics for real-time visualization of disk arm movement.

```
import matplotlib.pyplot as plt
```

- Generated a bar chart using Matplotlib to compare the total head movement of different algorithms.

## Results and Analysis

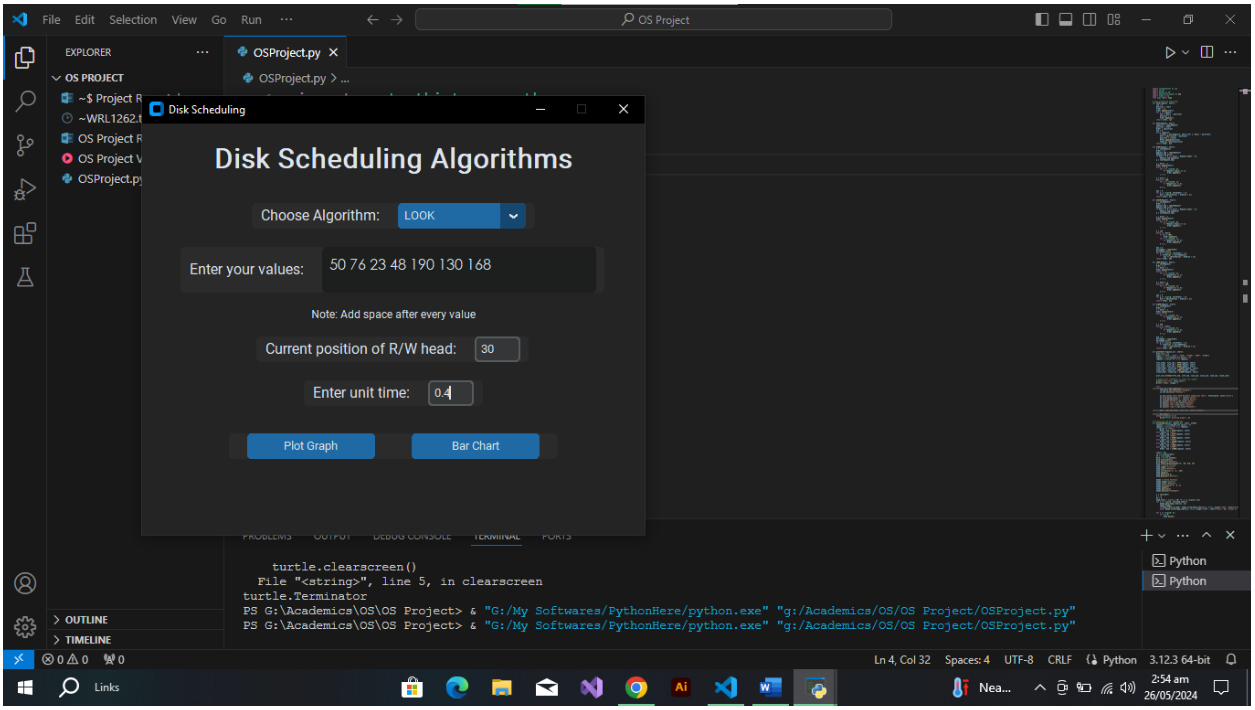
**Graphical Visualization:** Users can observe the movement of the disk arm in real-time and visualize how each algorithm handles disk requests.

**Bar Chart Comparison:** A bar chart displays the total head movement for each algorithm, aiding in comparative analysis and algorithm selection based on specific requirements.

## Conclusion

In conclusion, our project successfully implemented various disk scheduling algorithms and provided a user-friendly GUI for interactive visualization and analysis. By comparing the performance metrics of different algorithms, users can make informed decisions based on their system requirements and workload characteristics. Future enhancements may include additional scheduling algorithms and advanced visualization techniques for deeper analysis.

# Screenshots and Images



Disk Scheduling Algorithms

Choose Algorithm: LOOK

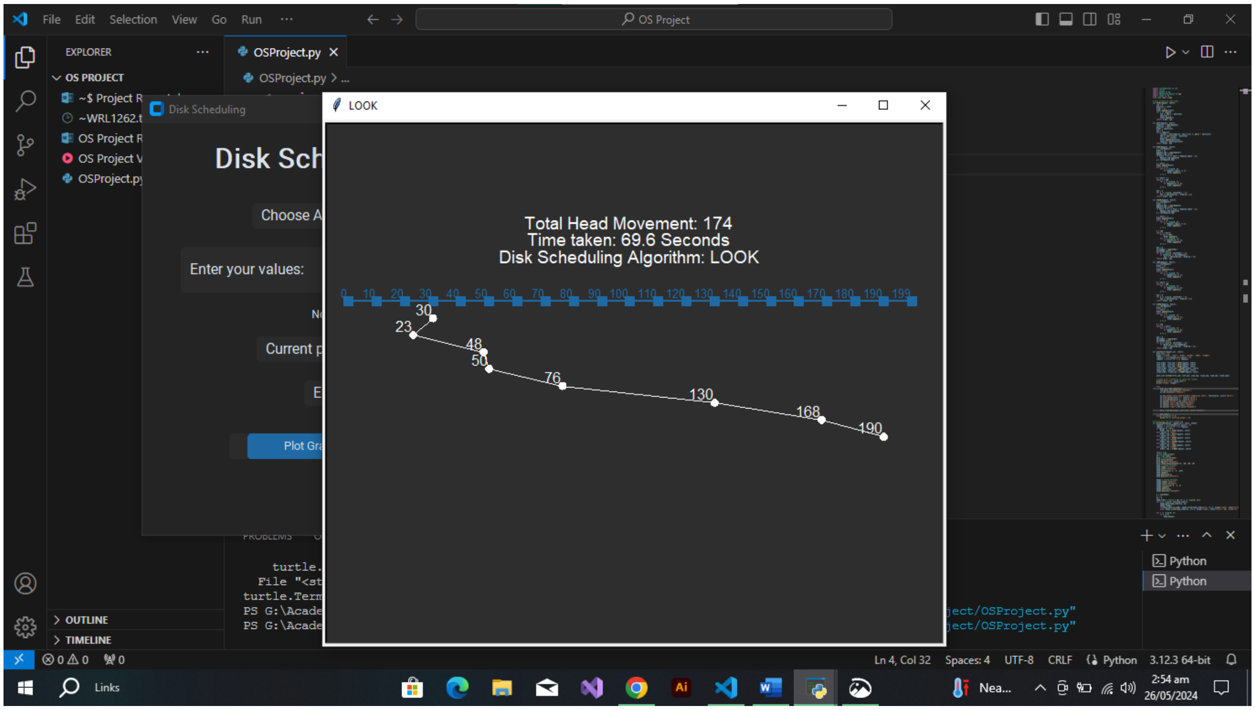
Enter your values: 50 76 23 48 190 130 168

Note: Add space after every value

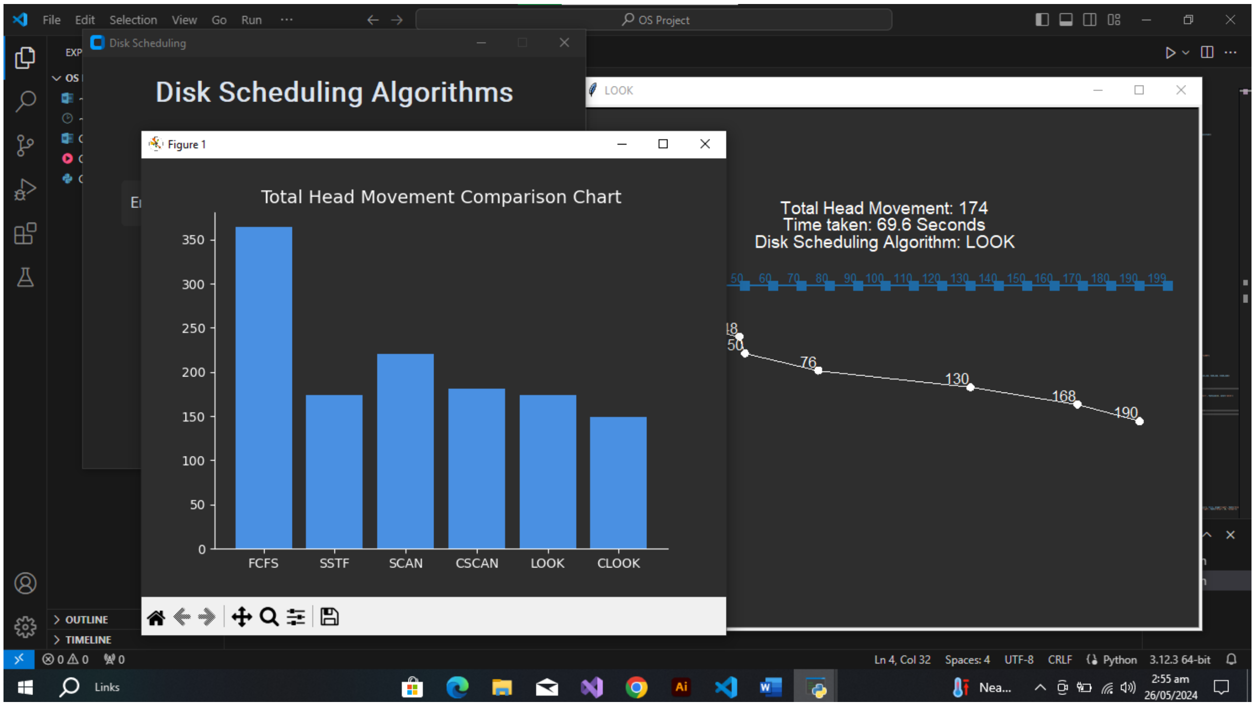
Current position of R/W head: 30

Enter unit time: 0.4

```
    turtle.clearscreen()
  File "<string>", line 5, in clearscreen
turtle.Terminator
PS G:\Academics\OS\OS Project> & "G:/My Softwares/PythonHere/python.exe" "g:/Academics/OS/OS Project/OSProject.py"
PS G:\Academics\OS\OS Project> & "G:/My Softwares/PythonHere/python.exe" "g:/Academics/OS/OS Project/OSProject.py"
```



Total Head Movement: 174
Time taken: 69.6 Seconds
Disk Scheduling Algorithm: LOOK



Disk Scheduling Algorithms

Total Head Movement: 174
Time taken: 69.6 Seconds
Disk Scheduling Algorithm: LOOK

## Video (Screen Recording of Project) Link

https://drive.google.com/drive/folders/10hQXtTMblxuFTcobuqyTCpJguBSr2KIV?usp=sharing

## Acknowledgements and References

We acknowledge the contributions of the open-source community and the libraries used in this project, including customtkinter, Turtle, Pandas, Matplotlib, and NumPy.

https://docs.python.org/3/library/turtle.html

https://pypi.org/project/customtkinter/0.3/

https://matplotlib.org/