---

1. FCFS Disk Scheduling

Program Name: FCFS Disk Scheduling

Sample Input:

Enter number of disk requests: 4

Enter the disk requests:

98 183 37 122

Enter the starting disk position: 53

Sample Output:

FCFS Disk Scheduling:

Move to 98

Move to 183

Move to 37

Move to 122

Total head movement: 276

---

2. SCAN Disk Scheduling

Program Name: SCAN Disk Scheduling

Sample Input:

Enter number of disk requests: 4

Enter the disk requests:

98 183 37 122

Enter the starting disk position: 53

Enter the direction (0 for left, 1 for right): 1

Enter the maximum disk position: 199

Sample Output:

SCAN Disk Scheduling:

Total head movement: 316

---

3. C-SCAN Disk Scheduling

Program Name: C-SCAN Disk Scheduling

Sample Input:

Enter number of disk requests: 4

Enter the disk requests:

98 183 37 122

Enter the starting disk position: 53

Enter the direction (0 for left, 1 for right): 1

Enter the maximum disk position: 199

Sample Output:

C-SCAN Disk Scheduling:

Total head movement: 366

---

4. One-Pass Assembler

Program Name: One-Pass Assembler

Sample Input (assembly.txt):

MOV AX

ADD BX

SUB CX

MUL DX

DIV AX

Sample Output:

Enter the assembly code file name: assembly.txt

Machine code for line 1: 02 00

Machine code for line 2: 00 01

Machine code for line 3: 01 02

Machine code for line 4: 03 03

Machine code for line 5: 04 00

---

5. Process Scheduling (FCFS, SJF, SRTF, Priority, Round Robin)

Program Name: Process Scheduling Algorithms (FCFS, SJF, SRTF, Priority, Round Robin)

Sample Input for FCFS:

Enter number of processes: 4

Enter the arrival times: 0 2 4 6

Enter the burst times: 5 3 8 6

Sample Output for FCFS:

Process Scheduling for FCFS:

Process 1 completed at time 5

Process 2 completed at time 8

Process 3 completed at time 16

Process 4 completed at time 22

Average Waiting Time: 4.0

Average Turnaround Time: 8.0

---

6. Deadlock Detection (Banker's Algorithm)

Program Name: Deadlock Detection using Banker's Algorithm

Sample Input:

Enter number of processes: 3

Enter number of resources: 3

Enter the available resources: 3 3 2

Enter the allocation matrix:

2 1 1

3 2 2

2 1 1

Enter the max matrix:

3 2 2

4 3 3

3 3 2

Sample Output:

Safe sequence exists: P0, P1, P2

System is in a safe state.

---

7. Paging Implementation (Memory Management)

Program Name: Paging Memory Management

Sample Input:

Enter the number of pages: 4

Enter the page size: 4

Enter the reference string:

7 0 1 2 0 3 0 4 5 6 7

Sample Output:

Page Faults: 8

Page Hits: 3

---

8. Segmentation Implementation (Memory Management)

Program Name: Segmentation Memory Management

Sample Input:

Enter the number of segments: 3

Enter the segment sizes:

100 200 150

Enter the logical addresses to translate:

50 220 120

Sample Output:

Segment 1, Offset 50 - Physical address 150

Segment 2, Offset 20 - Physical address 220

Segment 3, Offset 20 - Physical address 320

---

9. Disk Scheduling (FCFS, SCAN, and C-SCAN)

Program Name: Disk Scheduling Algorithms (FCFS, SCAN, C-SCAN)

Sample Input for FCFS:

Enter number of disk requests: 4

Enter the disk requests:

98 183 37 122

Enter the starting disk position: 53

Sample Output for FCFS:

FCFS Disk Scheduling:

Move to 98

Move to 183

Move to 37

Move to 122

Total head movement: 276

One-Pass Assembler

Sample Input (assembly.txt):

The assembly file (assembly.txt) contains the following instructions:

MOV AX

ADD BX

SUB CX

MUL DX

DIV AX

Program Execution:

Sample Input to the Program:

Enter the assembly code file name: assembly.txt

Sample Output:

The machine code output for each assembly instruction would look like the following:

Machine code for line 1: 02 00

Machine code for line 2: 00 01

Machine code for line 3: 01 02

Machine code for line 4: 03 03

Machine code for line 5: 04 00