

OS Assignment 8

Code:

```
#include <stdio.h>
#include <stdlib.h>

#define MAX 100

// SSTF (Shortest Seek Time First) Disk Scheduling
void sstf(int requests[], int n, int head) {
    int serviced[MAX], count = 0, totalMovement = 0;
    int visited[MAX] = {0};

    printf("\nSSTF Disk Scheduling\n");
    printf("Head Movement Sequence: %d", head);

    while (count < n) {
        int minDistance = 10000, minIndex;

        for (int i = 0; i < n; i++) {
            if (!visited[i]) {
                int distance = abs(requests[i] - head);
                if (distance < minDistance) {
                    minDistance = distance;
                    minIndex = i;
                }
            }
        }

        totalMovement += minDistance;
        head = requests[minIndex];
        visited[minIndex] = 1;
        serviced[count++] = head;

        printf(" -> %d", head);
    }

    printf("\nTotal Head Movement = %d\n", totalMovement);
}

// SCAN (Elevator) Disk Scheduling
void scan(int requests[], int n, int head, int diskSize, int direction) {
    int serviced[MAX], count = 0, totalMovement = 0;
    requests[n] = 0;
    requests[n + 1] = diskSize - 1;
    n += 2;
```

```

// Sort the requests
for (int i = 0; i < n - 1; i++) {
    for (int j = 0; j < n - i - 1; j++) {
        if (requests[j] > requests[j + 1]) {
            int temp = requests[j];
            requests[j] = requests[j + 1];
            requests[j + 1] = temp;
        }
    }
}

int position = 0;
for (int i = 0; i < n; i++) {
    if (requests[i] >= head) {
        position = i;
        break;
    }
}

printf("\nSCAN Disk Scheduling\n");
printf("Head Movement Sequence: %d", head);

if (direction == 1) { // Moving right
    for (int i = position; i < n; i++) {
        printf(" -> %d", requests[i]);
        totalMovement += abs(requests[i] - head);
        head = requests[i];
    }
    for (int i = position - 1; i >= 0; i--) {
        printf(" -> %d", requests[i]);
        totalMovement += abs(requests[i] - head);
        head = requests[i];
    }
} else { // Moving left
    for (int i = position - 1; i >= 0; i--) {
        printf(" -> %d", requests[i]);
        totalMovement += abs(requests[i] - head);
        head = requests[i];
    }
    for (int i = position; i < n; i++) {
        printf(" -> %d", requests[i]);
        totalMovement += abs(requests[i] - head);
        head = requests[i];
    }
}

printf("\nTotal Head Movement = %d\n", totalMovement);
}

```

```

// C-LOOK Disk Scheduling
void c_look(int requests[], int n, int head) {
    int totalMovement = 0;

    // Sort the requests
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (requests[j] > requests[j + 1]) {
                int temp = requests[j];
                requests[j] = requests[j + 1];
                requests[j + 1] = temp;
            }
        }
    }

    int position = 0;
    for (int i = 0; i < n; i++) {
        if (requests[i] >= head) {
            position = i;
            break;
        }
    }

    printf("\nC-LOOK Disk Scheduling\n");
    printf("Head Movement Sequence: %d", head);

    for (int i = position; i < n; i++) {
        printf(" -> %d", requests[i]);
        totalMovement += abs(requests[i] - head);
        head = requests[i];
    }

    for (int i = 0; i < position; i++) {
        printf(" -> %d", requests[i]);
        totalMovement += abs(requests[i] - head);
        head = requests[i];
    }

    printf("\nTotal Head Movement = %d\n", totalMovement);
}

int main() {
    int requests[MAX], n, head, diskSize, direction, choice;

    printf("Enter number of requests: ");
    scanf("%d", &n);

```

```

printf("Enter the requests: ");
for (int i = 0; i < n; i++) {
    scanf("%d", &requests[i]);
}

printf("Enter initial head position: ");
scanf("%d", &head);

printf("Enter disk size (for SCAN and C-LOOK): ");
scanf("%d", &diskSize);

while (1) {
    printf("\nChoose Disk Scheduling Algorithm:\n");
    printf("1. SSTF\n2. SCAN\n3. C-LOOK\n4. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch (choice) {
        case 1:
            sstf(requests, n, head);
            break;
        case 2:
            printf("Enter initial direction (1 for right, 0 for left): ");
            scanf("%d", &direction);
            scan(requests, n, head, diskSize, direction);
            break;
        case 3:
            c_look(requests, n, head);
            break;
        case 4:
            return 0;
        default:
            printf("Invalid choice!\n");
    }
}

return 0;
}

```

Output:

```

cd "c:\Users\Monika\Downloads\" ; if ($?) { gcc assignment8.c -o assignment8 } ; if ($?) {
.\assignment8 }
Enter number of requests: 7
Enter the requests: 34 54 90 58 121 79 31
Enter initial head position: 50
Enter disk size (for SCAN and C-LOOK): 200

```

Choose Disk Scheduling Algorithm:

1. SSTF
2. SCAN
3. C-LOOK
4. Exit

Enter your choice: 1

SSTF Disk Scheduling

Head Movement Sequence: 50 -> 54 -> 58 -> 79 -> 90 -> 121 -> 34 -> 31

Total Head Movement = 161

Choose Disk Scheduling Algorithm:

1. SSTF
2. SCAN
3. C-LOOK
4. Exit

Enter your choice: 2

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

SCAN Disk Scheduling

Head Movement Sequence: 50 -> 54 -> 58 -> 79 -> 90 -> 121 -> 199 -> 34 -> 31 -> 0

Total Head Movement = 348

Choose Disk Scheduling Algorithm:

1. SSTF
2. SCAN
3. C-LOOK
4. Exit

Enter your choice: 3

C-LOOK Disk Scheduling

Head Movement Sequence: 50 -> 54 -> 58 -> 79 -> 90 -> 0 -> 31 -> 34

Total Head Movement = 164

```
PS C:\Users\Monika\OneDrive\Desktop\pictoreal\Pictoreal-New-Main-Website> cd "c:\Users\Monika\Downloads\" ;
if ($?) { gcc assignment8.c -o assignment8 } ; if ($?) { .\assignment8 }
Enter number of requests: 7
Enter the requests: 34 54 90 58 121 79 31
Enter initial head position: 50
Enter disk size (for SCAN and C-LOOK): 200

Choose Disk Scheduling Algorithm:
1. SSTF
2. SCAN
3. C-LOOK
4. Exit
Enter your choice: 1

SSTF Disk Scheduling
Head Movement Sequence: 50 -> 54 -> 58 -> 79 -> 90 -> 121 -> 34 -> 31
Total Head Movement = 161

Choose Disk Scheduling Algorithm:
1. SSTF
2. SCAN
3. C-LOOK
4. Exit
Enter your choice: 2
Enter initial direction (1 for right, 0 for left): 1

SCAN Disk Scheduling
Head Movement Sequence: 50 -> 54 -> 58 -> 79 -> 90 -> 121 -> 199 -> 34 -> 31 -> 0
Total Head Movement = 348

Choose Disk Scheduling Algorithm:
1. SSTF
2. SCAN
3. C-LOOK
4. Exit
Enter your choice: 3

C-LOOK Disk Scheduling
Head Movement Sequence: 50 -> 54 -> 58 -> 79 -> 90 -> 0 -> 31 -> 34
Total Head Movement = 164
```