

OS Practice Assignment 4

1

Implement a C program to determine the optimal I/O order based on both Seek Time and Rotational Latency.

Define a struct Request containing int track and int sector.

Assumptions:

Seek Time = $\text{abs}(\text{current track} - \text{target track}) * 10\text{ms}$.

Rotational Latency = $\text{abs}(\text{current sector} - \text{target sector}) * 2\text{ms}$.

Task: Given a list of 5 requests and a current head position (track/sector), write a function to calculate the "Total Cost" for each request and sort/print them in the order they should be processed according to SPTF.

Implement a C program to determine the optimal I/O order based on both Seek Time and Rotational Latency.

2

Develop a C program that mimics the functionality of the `ls -li` command to inspect file metadata. The program must take a directory path as a command-line argument. Use `opendir()` to open the specified directory; Iterate through the directory entries using `readdir()`; For each file found, use the `stat()` system call to retrieve its metadata. Print the following details: File Name, File Size (bytes).

3

Implement a C program to manage free disk blocks using a Bitmap (Bit Array).

1. Define a global integer `disk_bitmap = 0`; (simulating a small disk with 32 blocks).
2. Implement a function `int allocate_block()` that uses bitwise operators to find the first bit that is 0, sets it to 1 (marking it as used), and returns the block index (0-31).
3. Implement a function `void free_block(int block_index)` that sets the corresponding bit back to 0.
4. Demonstrate allocating 3 blocks and freeing 1 block in the `main()` function, printing the bitmap status (in hex or binary) after each step.

4

Write a C program to simulate the behaviour of the SSTF disk scheduling algorithm.

Input:

Total number of tracks (e.g., 200).

Current position of the Read/Write Head (e.g., 50).

An array of pending requests: 82, 170, 43, 140, 24, 16, 190.

The head must always move to the closest pending request from its current position.

Output:

Print the sequence in which the requests are serviced.

Calculate and print the Total Head Movement (Total Seek Count).

5

Write a C program to count how many blocks are available (free) on a simulated disk.

Input: unsigned int disk_map = 0xF0F0; (Binary: 1111 0000 1111 0000). Assume the disk has 16 blocks (indices 0 to 15).

Use a loop to iterate through bits 0 to 15. Check each bit using bitwise operators. If the bit is 0, increment a counter.

Expected Output: Total Free Blocks: 8