Worst\_fit

#include <stdio.h>

#define MAX 10 // Maximum number of memory blocks and processes

int main() {

int blockSize[MAX], processSize[MAX], allocation[MAX];

int blocks, processes;

// Input number of memory blocks

printf("Enter number of memory blocks: ");

scanf("%d", &blocks);

// Input sizes of memory blocks

printf("Enter sizes of %d memory blocks:\n", blocks);

for (int i = 0; i < blocks; i++)

scanf("%d", &blockSize[i]);

// Input number of processes

printf("Enter number of processes: ");

scanf("%d", &processes);

// Input sizes of processes

printf("Enter sizes of %d processes:\n", processes);

for (int i = 0; i < processes; i++)

scanf("%d", &processSize[i]);

// Step 1: Initialize all processes as not allocated

for (int i = 0; i < processes; i++)

allocation[i] = -1;

// Step 2: Apply Worst Fit Allocation

for (int i = 0; i < processes; i++) {

int worstIdx = -1; // Index of the largest suitable block

// Check all blocks for worst fit

for (int j = 0; j < blocks; j++) {

if (blockSize[j] >= processSize[i]) {

// Choose the block with the maximum size

if (worstIdx == -1 || blockSize[j] > blockSize[worstIdx])

worstIdx = j;

}

}

// If a suitable block was found

if (worstIdx != -1) {

allocation[i] = worstIdx; // Assign block to process

blockSize[worstIdx] -= processSize[i]; // Reduce block size

}

}

// Step 3: Display allocation results

printf("\nProcess No.\tProcess Size\tBlock No.\n");

for (int i = 0; i < processes; i++) {

printf("%d\t\t%d\t\t", i + 1, processSize[i]);

if (allocation[i] != -1)

printf("%d\n", allocation[i] + 1); // +1 for human-readable block number

else

printf("Not Allocated\n");

}

return 0;

}

START

1. Input number of memory blocks → blocks

2. Input sizes of memory blocks → blockSize[]

3. Input number of processes → processes

4. Input sizes of processes → processSize[]

5. Set allocation[] = -1 for all processes

6. For each process i:

a. Set worstIdx = -1

b. For each block j:

i. If blockSize[j] ≥ processSize[i]:

- If worstIdx == -1 or blockSize[j] > blockSize[worstIdx]:

→ Set worstIdx = j

c. If worstIdx ≠ -1:

- allocation[i] = worstIdx

- blockSize[worstIdx] -= processSize[i]

7. Display: Process No, Process Size, Allocated Block No or Not Allocated

END