Best fit

#include <stdio.h>

#define MAX 10 // Maximum size for memory 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 block sizes

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 process sizes

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 Best Fit Allocation

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

int bestIdx = -1; // Index of best block for current process

// Check all blocks

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

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

// If this is the smallest suitable block found so far

if (bestIdx == -1 || blockSize[j] < blockSize[bestIdx])

bestIdx = j;

}

}

// If suitable block found

if (bestIdx != -1) {

allocation[i] = bestIdx; // Assign process to best block

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

}

}

// Step 3: Display the allocation result

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); // Add 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 each block → blockSize[]

3. Input number of processes → processes

4. Input sizes of each process → processSize[]

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

6. For each process i:

a. Initialize bestIdx = -1

b. For each block j:

i. If block j is large enough for process i:

- If bestIdx is -1 or block j < block[bestIdx], set bestIdx = j

c. If bestIdx ≠ -1:

- Assign process i to block bestIdx

- Reduce blockSize[bestIdx] by processSize[i]

7. For each process:

a. Print Process No, Process Size, Block No or Not Allocated

END