

22bps1059

one

```
File Edit View Terminal Tabs Help
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define MAX 5
4
5 void display(int *arr[MAX], int mems[MAX]) {
6     printf("Current memory\n");
7     for (int i = 0; i < MAX; i++) {
8         printf("Block %d: ", i);
9         for (int j = 0; j < mems[i]; j++) {
10             printf("%d ", arr[i][j]);
11         }
12         printf("\n");
13     }
14 }
15 int firstFit(int *arr[MAX], int mems[MAX], int request) {
16     printf("Allocating %d using First Fit\n", request);
17     for (int i = 0; i < MAX; i++) {
18         if (mems[i] >= request) {
19             for (int j = 0; j < request; j++) {
20                 arr[i][j] = 1;
21             }
22             return i;
23         }
24     }
25     return -1; // Allocation failed
26 }
27 int bestFit(int *arr[MAX], int mems[MAX], int request) {
28     printf("Allocating %d using Best Fit\n", request);
29     int minindex = -1;
30     for (int i = 0; i < MAX; i++) {
31         if (mems[i] >= request) {
32             if (minindex == -1 || mems[i] < mems[minindex]) {
33                 minindex = i;
34             }
35         }
36     }
37     if (minindex != -1) {
38         for (int j = 0; j < request; j++) {
39             arr[minindex][j] = 1;
40         }
41         return minindex;
42     }
43     return -1; // Allocation failed
44 }
45
46 int worstFit(int *arr[MAX], int mems[MAX], int request) {
47     printf("Allocating %d using Worst Fit\n", request);
48     int minindex = -1;
49     for (int i = 0; i < MAX; i++) {
50         if (mems[i] >= request) {
51             if (minindex == -1 || mems[i] > mems[minindex]) {
52                 minindex = i;
53             }
54         }
55     }
56     if (minindex != -1) {
57         for (int j = 0; j < request; j++) {
58             arr[minindex][j] = 1;
59         }
60         return minindex;
61     }
62     return -1; // Allocation failed
63 }
64
65 int main() {
66     int *arr[MAX] = {NULL};
67     int mems[MAX] = {0};
68
69     // Allocating memory
70     for (int i = 0; i < MAX; i++) {
71         int mem = rand() % 10 + 1;
72         mems[i] = mem;
73         printf("Allocating %d mem\n", mem);
74         arr[i] = calloc(mem, sizeof(int));
75     }
76
77     // Displaying memory
78     display(arr, mems);
79
80     int firstFitBlock = firstFit(arr, mems, 5);
81     if (firstFitBlock != -1) {
82         printf("Allocated in Block %d using First Fit\n", firstFitBlock);
83     } else {
84         printf("First Fit allocation failed\n");
85     }
86
87     int bestFitBlock = bestFit(arr, mems, 5);
88     if (bestFitBlock != -1) {
89         printf("Allocated in Block %d using Best Fit\n", bestFitBlock);
90     } else {
91         printf("Best Fit allocation failed\n");
92     }
93
94     int worstFitBlock = worstFit(arr, mems, 5);
95     if (worstFitBlock != -1) {
96         printf("Allocated in Block %d using Worst Fit\n", worstFitBlock);
97     } else {
98         printf("Worst Fit allocation failed\n");
99     }
100
101     display(arr, mems);
102
103     // Free allocated memory
104     for (int i = 0; i < MAX; i++) {
105         free(arr[i]);
106     }
107 }
108
cardi~/vit/os/lab10 ./one
Allocating 4 mem
Allocating 7 mem
Allocating 8 mem
Allocating 6 mem
Allocating 4 mem
Current memory
Block 0: 0 0 0 0
Block 1: 0 0 0 0 0 0
Block 2: 0 0 0 0 0 0 0
Block 3: 0 0 0 0 0 0
Block 4: 0 0 0 0
Allocating 5 using First Fit
Allocated in Block 1 using First Fit
Allocating 5 using Best Fit
Allocated in Block 3 using Best Fit
Allocating 5 using Worst Fit
Allocated in Block 2 using Worst Fit
Current memory
Block 0: 0 0 0 0
Block 1: 1 1 1 1 0 0
Block 2: 1 1 1 1 0 0 0
Block 3: 1 1 1 1 0
Block 4: 0 0 0 0
cardi~/vit/os/lab10 scrot --focused fig1.png
one.c 15,2 Top one.c [R0] 57,1 93%
[0] $!bash*
```

two

<pre> 1 #include <stdio.h> 2 #include <stdlib.h> 3 #include <stdbool.h> 4 #include <limits.h> 5 6 #define NUM_FRAMES 3 // Number of frames in memory 7 8 int frames[NUM_FRAMES]; // Memory frames 9 int referenceString[] = {7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2}; // Page refer 10 int n = sizeof(referenceString) / sizeof(referenceString[0]); 11 int pageFaults = 0; 12 // Initialize memory frames to -1 (empty) 13 void initializeFrames() { 14 for (int i = 0; i < NUM_FRAMES; i++) { 15 frames[i] = -1; 16 } 17 } 18 // Display the memory frames 19 void displayFrames() { 20 printf("Frames: "); 21 for (int i = 0; i < NUM_FRAMES; i++) { 22 if (frames[i] == -1) { 23 printf("Empty "); 24 } else { 25 printf("%d ", frames[i]); 26 } 27 } 28 printf("\n"); 29 } 30 // Check if a page is in memory frames 31 bool isPageInFrames(int page) { 32 for (int i = 0; i < NUM_FRAMES; i++) { 33 if (frames[i] == page) { 34 return true; 35 } 36 } 37 return false; 38 } 39 40 // FIFO Page Replacement Algorithm 41 void FIFO() { 42 initializeFrames(); 43 int front = 0; 44 45 printf("FIFO Page Replacement:\n"); 46 for (int i = 0; i < n; i++) { 47 int page = referenceString[i]; 48 if (!isPageInFrames(page)) { 49 pageFaults++; 50 frames[front] = page; 51 front = (front + 1) % NUM_FRAMES; </pre>	<pre> 71 if (frames[j] != -1) { 72 frames[j] = page; 73 break; 74 } 75 } 76 displayFrames(); 77 } 78 } 79 printf("Page Faults: %d\n", pageFaults); 80 } 81 82 // OPTIMAL Page Replacement Algorithm 83 void OPTIMAL() { 84 initializeFrames(); 85 86 printf("OPTIMAL Page Replacement:\n"); 87 for (int i = 0; i < n; i++) { 88 int page = referenceString[i]; 89 if (!isPageInFrames(page)) { 90 pageFaults++; 91 int farthest = -1; 92 int index = -1; 93 for (int j = 0; j < NUM_FRAMES; j++) { 94 int k; 95 for (k = i + 1; k < n; k++) { 96 if (frames[j] == referenceString[k]) { 97 if (k > farthest) { 98 farthest = k; 99 index = j; 100 } 101 } 102 } 103 if (k == n) { 104 index = j; 105 break; 106 } 107 } 108 frames[index] = page; 109 } 110 } 111 displayFrames(); 112 } 113 printf("Page Faults: %d\n", pageFaults); 114 } 115 116 int main() { 117 FIFO(); 118 pageFaults = 0; 119 MRU(); 120 pageFaults = 0; 121 OPTIMAL(); 122 } </pre>	<pre> cardi~/vit/os/lab10 ./two FIFO Page Replacement: Frames: 7 Empty Empty Frames: 7 0 Empty Frames: 7 0 1 Frames: 2 0 1 Frames: 2 0 1 Frames: 2 3 1 Frames: 2 3 0 Frames: 4 3 0 Frames: 4 2 0 Frames: 4 2 3 Frames: 0 2 3 Frames: 0 2 3 Frames: 0 2 3 Page Faults: 10 MRU Page Replacement: Frames: Empty Empty Empty Frames: Empty Empty Empty Frames: Empty Empty Empty Frames: Empty Empty Empty Frames: Empty Empty Empty Frames: Empty Empty Empty Frames: Empty Empty Empty Frames: Empty Empty Empty Frames: Empty Empty Empty Frames: Empty Empty Empty Page Faults: 13 OPTIMAL Page Replacement: Frames: 7 Empty Empty Frames: 0 Empty Empty Frames: 0 1 Empty Frames: 0 2 Empty Frames: 0 2 Empty Frames: 0 2 3 Frames: 0 2 3 Frames: 4 2 3 Frames: 4 2 3 Frames: 4 2 3 Frames: 0 2 3 Frames: 0 2 3 Frames: 0 2 3 Page Faults: 7 cardi~/vit/os/lab10 scrot --focused fig2.png </pre>
---	---	---