

```

3- for (i = 0; i < 50; i++) {
4-     f[i] = 0;
5- }
6-
7- do {
8-     count = 0;
9-     printf("Enter starting block and length of files: ");
10-    scanf("%d%d", &st, &len);
11-
12-    // Check if all required blocks are free
13-    for (k = st; k < (st + len); k++) {
14-        if (k >= 50) { // Ensure the index does not go out of bounds
15-            printf("Error: Out of bounds. Allocation failed.\n");
16-            count = -1;
17-            break;
18-        }
19-        if (f[k] == 0) {
20-            count++;
21-        }
22-    }
23-
24-    if (count == len) {
25-        // Allocate blocks
26-        for (j = st; j < (st + len); j++) {
27-            f[j] = 1;
28-            printf("%d\t%d\n", j, f[j]);
29-        }
30-        printf("The file is allocated to disk.\n");
31-    } else if (count != -1) {
32-        printf("The file is not allocated (insufficient contiguous space).\n");
33-    }
34-
35-    printf("Do you want to enter more files? (Yes - 1 / No - 0): ");
36-    scanf("%d", &c);
37-
38- } while (c == 1);
39-
40- return 0;
41- }

```

```

- Enter starting block and length of files: 5 3
5 1
6 1
7 1
The file is allocated to disk.
Do you want to enter more files? (Yes - 1 / No - 0): 1
Enter starting block and length of files: 6 4
The file is not allocated (insufficient contiguous space).
Do you want to enter more files? (Yes - 1 / No - 0): 0

=== Code Execution Successful ===

```

```

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

int main() {
    int RQ[100], i, n, TotalHeadMovement = 0, initial, max_cylinder;
    printf("Enter the maximum cylinder value on the disk: ");
    scanf("%d", &max_cylinder);
    printf("Enter the number of requests: ");
    scanf("%d", &n);
    if (n <= 0 || n > 100) {
        printf("Error: Invalid number of requests. Must be between 1 and 100.\n");
        return 1;
    }

    printf("Enter the request sequence (values between 0 and %d):\n", max_cylinder);
    for (i = 0; i < n; i++) {
        scanf("%d", &RQ[i]);
        if (RQ[i] < 0 || RQ[i] > max_cylinder) {
            printf("Error: Request %d is out of bounds. Must be between 0 and %d.\n", RQ[i], max_cylinder);
            return 1;
        }
    }

    printf("Enter initial head position (between 0 and %d): ", max_cylinder);
    scanf("%d", &initial);

    if (initial < 0 || initial > max_cylinder) {
        printf("Error: Initial head position out of bounds. Must be between 0 and %d.\n", max_cylinder);
        return 1;
    }

    for (i = 0; i < n; i++) {
        TotalHeadMovement += abs(RQ[i] - initial);
        initial = RQ[i];
    }

    printf("Total head movement is %d\n", TotalHeadMovement);
    return 0;
}

```

```

Enter the maximum cylinder value on the disk: 199
Enter the number of requests: 5
Enter the request sequence (values between 0 and 199):
45 80 120 10 90
Enter initial head position (between 0 and 199): 50
Total head movement is 270

```

```

=== Code Execution Successful ===

```

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4     int n, head, direction, total_head_movement = 0, max_cylinder;
5     printf("Enter the maximum cylinder value on the disk: ");
6     scanf("%d", &max_cylinder);
7     printf("Enter the number of disk requests: ");
8     scanf("%d", &n);
9     if (n <= 0 || n > 100) {
10         printf("Error: Invalid number of requests. Must be between 1 and 100.\n");
11         return 1;
12     }
13     int requests[n + 1];
14     printf("Enter the disk requests (values between 0 and %d):\n", max_cylinder);
15     for (int i = 0; i < n; i++) {
16         scanf("%d", &requests[i]);
17         if (requests[i] < 0 || requests[i] > max_cylinder) {
18             printf("Error: Request %d is out of bounds.\n", requests[i]);
19             return 1;
20         }
21     }
22     printf("Enter the initial head position (between 0 and %d): ", max_cylinder);
23     scanf("%d", &head);
24     if (head < 0 || head > max_cylinder) {
25         printf("Error: Initial head position out of bounds.\n");
26         return 1;
27     }
28     printf("Enter the head movement direction (1 for right, 0 for left): ");
29     scanf("%d", &direction);
30     requests[n] = (direction == 1) ? max_cylinder : 0;
31     n++;
32     for (int i = 0; i < n - 1; i++) {
33         for (int j = i + 1; j < n; j++) {
34             if (requests[i] > requests[j]) {
35                 int temp = requests[i];
36                 requests[i] = requests[j];
37                 requests[j] = temp;
38             }
39         }
40     }

```

```

- Enter the maximum cylinder value on the disk: 199
Enter the number of disk requests: 8
Enter the disk requests (values between 0 and 199):
98 183 37 122 14 124 65 67
Enter the initial head position (between 0 and 199): 53
Enter the head movement direction (1 for right, 0 for left): 1
Total head movement is: 331

```

=== Code Execution Successful ===

```

    }
}
int index = 0;
for (int i = 0; i < n; i++) {
    if (requests[i] >= head) {
        index = i;
        break;
    }
}
if (direction == 1) {
    for (int i = index; i < n; i++) {
        total_head_movement += abs(requests[i] - head);
        head = requests[i];
    }
    for (int i = index - 1; i >= 0; i--) {
        total_head_movement += abs(requests[i] - head);
        head = requests[i];
    }
} else {
    for (int i = index - 1; i >= 0; i--) {
        total_head_movement += abs(requests[i] - head);
        head = requests[i];
    }
    for (int i = index; i < n; i++) {
        total_head_movement += abs(requests[i] - head);
        head = requests[i];
    }
}
printf("Total head movement is: %d\n", total_head_movement);
return 0;
}

```

Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

GmailYouTubeMapsSSE ARMS LOGIN

All Bookmarks

Programiz
C Online Compiler

ANDTRONICS 4 Port USB 2.0 KVM Switch Box with 4 KVM Cables to Control u...
\$1,999.00 \$2,500.00 prime
Save 25% with coupon

Programiz PRO

main.c

```
31 for (j = 0; j < frames; j++) {
32     if (of[j] == refString[i]) {
33         flag1 = flag2 = 1;
34         time[j] = ++count;
35         break;
36     }
37 }
38 if (flag1 == 0) {
39     for (j = 0; j < frames; j++) {
40         if (m[j] == -1) {
41             m[j] = refString[i];
42             time[j] = ++count;
43             pageFaults++;
44             flag2 = 1;
45             break;
46         }
47     }
48 }
49 if (flag2 == 0) {
50     pos = findLeastRecentlyUsed(time, frames);
51     m[pos] = refString[i];
52     time[pos] = ++count;
53     pageFaults++;
54 }
55 for (k = 0; k < frames; k++) {
56     if (m[k] != -1) {
57         printf("%d\t", m[k]);
58     } else {
59         printf("\t");
60     }
61 }
62 if (flag1 == 0) {
63     printf("PF No. %d", pageFaults);
64 }
65 printf("\n");
66 }
67 printf("\nThe number of Page Faults using LRU are: %d\n", pageFaults);
68 return 0;
69 }
70 }
```

Run

Share

Clear

Output

Enter the number of frames: 3
Enter the length of the reference string: 12
Enter the reference string: 7 0 1 2 0 3 0 4 2 3 0 3

The Page Replacement Process is:
7 - - PF No. 1
7 0 - PF No. 2
7 0 1 PF No. 3
2 0 1 PF No. 4
2 0 1
2 0 3 PF No. 5
2 0 3
4 0 3 PF No. 6
4 0 2 PF No. 7
4 3 2 PF No. 8
0 3 2 PF No. 9
0 3 2

The number of Page Faults using LRU are: 9

--- Code Execution Successful ---

78°F
Partly sunny

Search

ENG
IN

09:38
28-12-2024

Programiz PRO

Premium
Courses by
Programiz

Learn More

Sania0906/os x Google Passwor x Download histor x 1-10 (1).pdf x (1) WhatsApp x 31-40 (1).pdf x ChatGPT x Online C Compil x

programiz.com/c-programming/online-compiler/

Gmail YouTube Maps SSE ARMS LOGIN

All Bookmarks

Programiz
C Online Compiler

amazon prime

Mile itna kuch ki lage
Sach Mein Too Much!

Join Prime Now

Programiz PRO

main.c

30 for (i = 0; i < totalReferences; i++) {
31 scanf("%d", &refString[i]);
32 }
33 for (i = 0; i < frameCount; i++) {
34 frames[i] = -1;
35 }
36 printf("\nThe Page Replacement Process is:\n");
37 for (i = 0; i < totalReferences; i++) {
38 int found = 0;
39 for (j = 0; j < frameCount; j++) {
40 if (frames[j] == refString[i]) {
41 found = 1;
42 break;
43 }
44 }
45 if (!found) {
46 pageFaults++;
47 if (i < frameCount) {
48 frames[i] = refString[i];
49 } else {
50 int pos = findOptimal(refString, frames, i, totalReferences, frameCount);
51 frames[pos] = refString[i];
52 }
53 }
54 for (k = 0; k < frameCount; k++) {
55 if (frames[k] != -1) {
56 printf("%d\t", frames[k]);
57 } else {
58 printf("\t");
59 }
60 }
61 if (!found) {
62 printf("PF No. %d", pageFaults);
63 }
64 printf("\n");
65 }
66 printf("\nThe number of Page Faults using Optimal Page Replacement are: %d\n", pageFaults);
67 return 0;
68 }
69 }

Run

Share

Clear

Output

Enter the number of frames: 3
Enter the length of the reference string: 12
Enter the reference string: 7 0 1 2 0 3 0 4 2 3 0 3

The Page Replacement Process is:
7 - - PF No. 1
7 0 - PF No. 2
7 0 1 PF No. 3
2 0 1 PF No. 4
2 0 1
2 0 3 PF No. 5
2 0 3
2 4 3 PF No. 6
2 4 3
2 4 3
0 4 3 PF No. 7
0 4 3

The number of Page Faults using Optimal Page Replacement are: 7

--- Code Execution Successful ---

Programiz PRO

Premium Courses by Programiz
Learn More

78°F
Partly sunny

Search

ENG IN

09:42
28-12-2024

Sania0906/os xGoogle Passw xDownload h... x1-10 (1).pdf x(1) WhatsApp x31-40 (1).pdf xChatGPT xOnline C Co xMicrosoft Co x+

programiz.com/c-programming/online-compiler/

GmailYouTubeMapsSSE ARMS LOGINAll Bookmarks

ProgramizC Online Compiler

Download Chrome Today - Install Google Chrome Today - Use Chrome for Faster Browsing

Programiz PRO

main.c

1#include <stdio.h>
2#include <stdlib.h>
3
4int main() {
5 int f[50] = {0}; // File allocation table, initialized to 0
6 int index[50]; // Array to store blocks pointed by the index block
7 int i, n, st, len, j, c, k, ind, count;
8
9 do {
10 printf("Enter the index block: ");
11 scanf("%d", &ind);
12
13 // Check if the index block is already allocated
14 if (f[ind] == 1) {
15 printf("Error: %d index is already allocated.\n", ind);
16 continue;
17 }
18
19 printf("Enter the number of blocks needed for the index %d on the disk: ", ind);
20 scanf("%d", &n);
21
22 count = 0;
23 printf("Enter the blocks for the index block: ");
24 for (i = 0; i < n; i++) {
25 scanf("%d", &index[i]);
26
27 // Check if the block is available
28 if (index[i] < 0 || index[i] >= 50) {
29 printf("Error: Block %d is out of bounds. Allocation failed.\n", index[i]);
30 break;
31 }
32 if (f[index[i]] == 0) {
33 count++;
34 }
35 }
36 if (count == n) {
37 f[ind] = 1;
38 for (j = 0; j < n; j++) {
39 f[index[j]] = 1;
40 }
41 printf("Allocation Successful\n");
42 }
43 } while (1);
44 }

Output

Enter the index block: 5
Enter the number of blocks needed for the index 5 on the disk: 3
Enter the blocks for the index block: 10 11 12
Allocated
File Indexed:
5 -----> 10 : 1
5 -----> 11 : 1
5 -----> 12 : 1
Do you want to enter more files? (Yes - 1 / No - 0): 1
Enter the index block: 10
Error!
Error: 10 index is already allocated.
Enter the index block: 6
Enter the number of blocks needed for the index 6 on the disk: 2
Enter the blocks for the index block: 12 13ERROR!

Error: Some blocks are already allocated or out of bounds. Allocation failed.
Do you want to enter more files? (Yes - 1 / No - 0): 0

--- Code Execution Successful ---

78°F
Partly sunny

Search

ENG
IN

09:52
28-12-2024

Programiz PRO
Premium
Courses by
Programiz
Learn More

Sania0906/os xGoogle Passw xDownload hi x1-10 (1).pdf x(1) WhatsApp x31-40 (1).pdf xChatGPT xOnline C Co xMicrosoft Co x+

programiz.com/c-programming/online-compiler/

GmailYouTubeMapsSSE ARMS LOGINAll Bookmarks

ProgramizC Online Compiler

Download Chrome Today - Install Google Chrome Today - Use Chrome for Faster Browsing

Programiz PRO

main.c

15printf("Error: %d index is already allocated.\n", ind);
16continue;
17}
18
19printf("Enter the number of blocks needed for the index %d on the disk: ", ind);
20scanf("%d", &n);
21
22count = 0;
23printf("Enter the blocks for the index blocks: ");
24for (i = 0; i < n; i++) {
25scanf("%d", &index[i]);
26
27// Check if the block is available
28if (index[i] < 0 || index[i] >= 50) {
29printf("Error: Block %d is out of bounds. Allocation failed.\n", index[i]);
30break;
31}
32if (f[index[i]] == 0) {
33count++;
34}
35}
36if (count == n) {
37f[ind] = 1;
38for (j = 0; j < n; j++) {
39f[index[j]] = 1;
40}
41printf("Allocated\n");
42printf("File Indexed\n");
43for (k = 0; k < n; k++) {
44printf("%d -----> %d : %d\n", ind, index[k], f[index[k]]);
45}
46}
47printf("Error: Some blocks are already allocated or out of bounds. Allocation failed.\n");
48}
49printf("Do you want to enter more files? (Yes - 1 / No - 0): ");
50scanf("%d", &c);
51}
52while (c == 1);
53return 0;
54}

Run

Output

Enter the index block: 5
Enter the number of blocks needed for the index 5 on the disk: 3
Enter the blocks for the index block: 10 11 12
Allocated
File Indexed:
5 -----> 10 : 1
5 -----> 11 : 1
5 -----> 12 : 1
Do you want to enter more files? (Yes - 1 / No - 0): 1
Enter the index block: 10
ERROR!
Error: 10 index is already allocated.
Enter the index block: 6
Enter the number of blocks needed for the index 6 on the disk: 2
Enter the blocks for the index block: 12 13ERROR!

Error: Some blocks are already allocated or out of bounds. Allocation failed.
Do you want to enter more files? (Yes - 1 / No - 0): 0

--- Code Execution Successful ---

Clear

Programiz PRO

Premium Courses by Programiz
Learn More

78°F
Partly sunny

Search

ENG
IN

09:52
28-12-2024

Sania0906/os xGoogle PassxDownload hix1-10 (1).pdf x(1) WhatsApp x31-40 (1).pdf xChatGPT xOnline C ConxMicrosoft Co x+

programiz.com/c-programming/online-compiler/

GmailYouTubeMapsSSE ARMS LOGINAll Bookmarks

ProgramizC Online Compiler

amazon primeMile itna kuch ki lageSach Mein Too Much!Join Prime NowProgramiz PRO

main.c

1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4 int f[50] = {0};
5 int p, i, st, len, j, c, a, success;
6 printf("Enter how many blocks are already allocated: ");
7 scanf("%d", &p);
8
9 printf("Enter the blocks already allocated: ");
10 for (i = 0; i < p; i++) {
11 scanf("%d", &a);
12 if (a >= 0 && a < 50) {
13 f[a] = 1; // Mark block as allocated
14 } else {
15 printf("Error: Block %d is out of bounds.\n", a);
16 }
17 }
18 do {
19 success = 1;
20 printf("Enter starting block and length: ");
21 scanf("%d%d", &st, &len);
22 if (st < 0 || st + len > 50) {
23 printf("Error: Block range is out of bounds.\n");
24 success = 0;
25 } else {
26 for (j = st; j < st + len; j++) {
27 if (f[j] == 1) {
28 printf("Error: Block %d is already allocated.\n", j);
29 success = 0;
30 break;
31 }
32 }
33 }
34 if (success) {
35 for (j = st; j < st + len; j++) {
36 f[j] = 1;
37 printf("%d -----> Allocated\n", j);
38 }
39 } else {
40 printf("Allocation failed.\n");
41 }
42 } while (1);
43 }

Output

Enter how many blocks are already allocated: 2
Enter the blocks already allocated: 5 10
Enter starting block and length: 7 3
7 -----> Allocated
8 -----> Allocated
9 -----> Allocated
Do you want to enter more files? (Yes - 1 / No - 0): 1
Enter starting block and length: 5 4
ERROR!
Error: Block 5 is already allocated.
Allocation failed.
Do you want to enter more files? (Yes - 1 / No - 0): 0

--- Code Execution Successful ---

78°FPartly sunny

Search

ENG IN09:5428-12-2024

Programiz PROPremium Courses by ProgramizLearn More

Sania0906/os xGoogle PassxDownload hix1-10 (1).pdf x(1) WhatsApp x31-40 (1).pdf xChatGPT xOnline C ConxMicrosoft Co x+

programiz.com/c-programming/online-compiler/

GmailYouTubeMapsSSE ARMS LOGINAll Bookmarks

Programiz
C Online Compiler

amazon prime

Mile itna kuch ki lage
Sach Mein Too Much!

Join Prime Now

Programiz PRO

main.c

891011121314151617181920212223242526272829303132333435363738394041424344454647

```
printf("Enter the blocks already allocated: ");
for (i = 0; i < p; i++) {
    scanf("%d", &a);
    if (a == 0 && a < 50) {
        t[a] = 1; // Mark block as allocated
    } else {
        printf("Error: Block %d is out of bounds.\n", a);
    }
}

do {
    success = 1;
    printf("Enter starting block and length: ");
    scanf("%d%d", &st, &len);
    if (st < 0 || st + len > 50) {
        printf("Error: Block range is out of bounds.\n");
        success = 0;
    } else {
        for (j = st; j < st + len; j++) {
            if (t[j] == 1) {
                printf("Error: Block %d is already allocated.\n", j);
                success = 0;
                break;
            }
        }
    }
    if (success) {
        for (j = st; j < st + len; j++) {
            t[j] = 1;
            printf("%d -----> Allocated\n", j);
        }
    } else {
        printf("Allocation failed.\n");
    }
    printf("Do you want to enter more files? (Yes - 1 / No - 0): ");
    scanf("%d", &c);
} while (c == 1);
return 0;
}
```

Run

Output

Enter how many blocks are already allocated: 2
Enter the blocks already allocated: 5 10
Enter starting block and length: 7 3
7 -----> Allocated
8 -----> Allocated
9 -----> Allocated
Do you want to enter more files? (Yes - 1 / No - 0): 1
Enter starting block and length: 5 4
ERROR!
Error: Block 5 is already allocated.
Allocation failed.
Do you want to enter more files? (Yes - 1 / No - 0): 0

--- Code Execution Successful ---

Clear

78°F
Partly sunny

Search

ENG
IN

09:54
28-12-2024