

Open



Save



A1c

Assignment 4

A1.c

setc.c

```
1|
2 #include<stdio.h>
3
4
5 void showPaging(int pageFrame[], int n){
6     int i=0;
7     for(i=0; i<n; i++){
8         printf("| %d ", pageFrame[i]);
9     }
10    printf("\n");
11}
12
13
14 void FIFO(int refString[], int n, int len){
15     int pageFrame[n];
16     int i,j;
17     int page, isPresent;
18     int replaceIndex=0;
19     int pageFaults = 0;
20
21     for(i=0; i<n; i++){
22         pageFrame[i] = 0;
23     }
24
25     for(i=0; i<len ; i++){
26         page = refString[i];
27         printf("\npage : %d\n", page);
28         isPresent = 0;
29         for(j=0; j<=n; j++){
30             if(pageFrame[j] == page){
31                 isPresent = 1;
32                 break;
33             }
34         }
35     }
}
```

Open



Save



A1.c
~/OS1/Assignment 4

A1c

```
setc.c x A1c
25     for(i=0; i<len; i++){
26         page = refString[i];
27         printf("\npage : %d\n", page);
28         isPresent = 0;
29         for(j=0; j<n; j++){
30             if(pageFrame[j] == page){
31                 isPresent = 1;
32                 break;
33             }
34         }
35         if(!isPresent){
36             pageFrame[replaceIndex % n] = refString[i];
37             replaceIndex++; |
38             pageFaults++;
39             showPaging(pageFrame, n);
40         }
41         else {
42             continue;
43         }
44     }
45 }
46
47 printf("\nPage faults : %d\n", pageFaults);
48
49 }
50
51
52 int findReplaceIndex(int recUsed[], int n){
53     int i=0, min=recUsed[0], index=0;
54     for(i=1; i<n; i++){
55         if(min > recUsed[i]){
56             min = recUsed[i];
57             index = i;
58         }
59     }
60 }
```

C ▾ Tab Width: 4 ▾

Ln 1, Col 1

INS



```
58     }
59 }
60     return index;
61 }
62
63 void LRU(int refString[], int n, int len){
64     int pageFrame[n];
65     int recUsed[n];
66
67     int i,j;
68     int page, isPresent;
69
70     int replaceIndex=0;           █
71     int pageFaults = 0;
72
73     // initialize arrays
74     for(i=0; i<n; i++){
75         pageFrame[i] = 0;
76         recUsed[i] = -1;
77     }
78
79     for(i=0; i<len; i++){
80         page = refString[i];
81         printf("\npage : %d\n", page);
82         isPresent = 0;
83
84         for(j=0; j<n; j++){
85             if(pageFrame[j] == page){
86                 isPresent = 1;
87                 break;
88             }
89         }
90
91         if(!isPresent){
92             replaceIndex = findReplaceIndex(recUsed, n);
```

C ▾ Tab Width: 4 ▾

Ln 1, Col 1

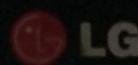
INS

```
86         isPresent = 1;
87     }
88 }
89
90 if(!isPresent){
91     replaceIndex = findReplaceIndex(recUsed, n);
92     //printf("\nreplaceindex : %d \n", replaceIndex);
93     pageFrame[replaceIndex] = refString[i];
94     recUsed[replaceIndex] = i;
95
96     showPaging(pageFrame, n);
97     pageFaults++;
98 }
99 else {
100     recUsed[j] = i;
101 }
102
103 }
104
105 printf("\nPage faults : %d\n", pageFaults);
106 }
107
108 void main() {
109     int refString[] = {12,15,12,18,6,8,11,12,19,12,6,8,12,15,19,8};
110     int n;
111     printf("Enter the number of memory refString (n): ");
112     scanf("%d", &n);
113
114     printf("\n***** FIFO *****\n");
115     FIFO(refString, n, 16);
116
117     printf("\n***** LRU *****\n");
118     LRU(refString, n, 16);
119 }
120 }
```

C ▾ Tab Width: 4 ▾

Ln 1, Col 1

INS



```
1
2 #include<stdio.h>
3
4
5 void showPaging(int pageFrame[], int n){
6     int i=0;
7     for(i=0; i<n; i++){
8         printf(" | %d ", pageFrame[i]);
9     }
10    printf("\n");
11 }
12
13
14 int findMinReplaceIndex(int recUsed[], int n){
15     int i=0, min=recUsed[0], index=0;
16     for(i=1; i<n; i++){
17         if(min > recUsed[i]){
18             min = recUsed[i];
19             index = i;
20         }
21     }
22     return index;
23 }
24
25 void LFU(int refString[], int n, int len){
26     int pageFrame[n];
27     int recUsed[n];
28
29     int i,j;
30     int page, isPresent;
31
32     int replaceIndex=0;
33     int pageFaults = 0;
34
35     // initialize arrays
```

Open



setc.c

~ /OS1/Assignment 4

setc

x

A1c

```
33     int pageFaults = 0;
34
35     // initialize arrays
36     for(i=0; i<n; i++){
37         pageFrame[i] = 0;
38         recUsed[i] = -1;
39     }
40
41     for(i=0; i<len; i++){
42         page = refString[i];
43         printf("\npage : %d\n", page);
44         isPresent = 0;
45
46         for(j=0; j<n; j++){
47             if(pageFrame[j] == page){
48                 isPresent = 1;
49                 break;
50             }
51         }
52
53         if(!isPresent){
54             replaceIndex = findMinReplaceIndex(recUsed, n);
55             //printf("\nreplaceIndex : %d \n", replaceIndex);
56             pageFrame[replaceIndex] = refString[i];
57             recUsed[replaceIndex] += 1;
58
59             showPaging(pageFrame, n);
60             pageFaults++;
61         }
62         else {
63             recUsed[j] += 1;
64         }
65     }
66
67     printf("\nPage faults : %d\n", pageFaults);
```

C ▾ Tab Width: 4 ▾

Ln 1, Col 1

Open



setc.c

~/OS1/Assignment 4

setc.c



A1c

```
78
71 int findMaxReplaceIndex(int recUsed[], int n){
72     int i=0, max=-100, index=0;
73     for(i=0; i<n; i++){
74         if(recUsed[i] > max){
75             max = recUsed[i];
76             index = i;
77         }
78     }
79     return index;
80 }
81
82
83
84 void MRU(int refString[], int n, int len){
85     int pageFrame[n];
86     int recUsed[n];
87
88     int i,j,l=0;
89     int page, isPresent;
90
91     int replaceIndex=0;
92     int pageFaults = 0;
93
94     // initialize arrays
95     for(i=0; i<n; i++){
96         pageFrame[i] = 0;
97         recUsed[i] = 0;
98     }
99
100    for(i=0; i<len; i++){
101        page = refString[i];
102        printf("\npage : %d\n", page);
103        isPresent = 0;
```

Open



Save

A1.c

```
99
100    for(i=0; i<len; i++){
101        page = refString[i];
102        printf("\npage : %d\n", page);
103        isPresent = 0;
104
105        for(j=0; j<n; j++){
106            if(pageFrame[j] == page){
107                isPresent = 1;
108                break;
109            }
110        }
111
112        if(!isPresent){
113            if(l<n){
114                pageFrame[l] = refString[i];
115                recUsed[l] = i;
116                l++;
117            }
118            else{
119                replaceIndex = findMaxReplaceIndex(recUsed, n);
120                pageFrame[replaceIndex] = refString[i];
121                recUsed[replaceIndex] = i;
122            }
123            showPaging(pageFrame, n);
124            pageFaults++;
125        }
126        else {
127            recUsed[j] = i;
128        }
129    }
130
131    printf("\nPage faults : %d\n", pageFaults);
132 }
```

Open



setc.c

~/OS1/Assignment 4

Save

A1.c

```
110 }
111
112     if(!isPresent){
113         if(l<n){
114             pageFrame[l] = refString[i];
115             recUsed[l] = i;
116             l++;
117         }
118         else{
119             replaceIndex = findMaxReplaceIndex(recUsed, n);
120             pageFrame[replaceIndex] = refString[i];
121             recUsed[replaceIndex] = i;
122         }
123         showPaging(pageFrame, n);
124         pageFaults++;
125     }
126     else {
127         recUsed[j] = i;
128     }
129 }
130
131 printf("\nPage faults : %d\n", pageFaults);
132 }
133
134
135 void main() {
136 // int refString[] = {12,15,12,18,6,8,11,12,19,12,6,8,12,15,19,8};
137 int refString[] = {2,5,2,8,5,4,1,2,3,2,6,1,2,5,9,8};
138 int n;
139 printf("Enter the number of memory refString (n): ");
140 scanf("%d", &n);
141 printf("\n***** LFU *****\n");
142 //LFU(refString, n, 16);
143 MRU(refString, n, 16);
144 }
```