

```

1  #include <Windows.h>
2  #include <stdio.h>
3
4  int main(void) {
5      //shared lock
6      HANDLE hFileHandle;
7      INT EXCLUSIVE = 0, SHARED = 1;
8      BOOL bSuccess;
9      char *szFilename = "testing.txt";
10     char *szFile = "test1.txt";
11     char szBuffer[245];
12
13     ///file uusgeh
14     hFileHandle = CreateFile(szFilename, GENERIC_READ,
15         FILE_SHARE_READ, 0, CREATE_ALWAYS, 0, 0);
16     if(hFileHandle == INVALID_HANDLE_VALUE) {
17         printf("Shine file uusgei chdsangui.\n");
18         exit(EXIT_FAILURE);
19     }
20     FILE *fptr = fopen("testing.txt", "rb");
21     if(fptr == NULL)
22     {
23         printf("Error!");
24         exit(1);
25     }
26     fprintf(fptr, "%d", 1234);
27     fseek(fptr, 0L, SEEK_END);
28     int lengthOfFile = ftell(fptr);
29     fclose(fptr);
30
31     ///bufferiin urtaar file lock hiine. Buten lock
32     bSuccess = LockFile(hFileHandle, 0, 0, lengthOfFile, 0);
33     if(bSuccess) {
34         printf("\n\nshared: File lock on %s secured.\n", szFilename);
35     }
36     bSuccess = UnlockFile(hFileHandle, 0, 0, lengthOfFile, 0);
37     if(bSuccess) {
38         printf("shared: File lock on %s released.\n", szFilename);
39     }
40
41     ///hagas lock
42     bSuccess = LockFile(hFileHandle, 0, 0, lengthOfFile/2, 0);
43     if(bSuccess) {
44         printf("shared: File half lock on %s secured.\n", szFilename);
45     }
46
47     bSuccess = UnlockFile(hFileHandle, 0, 0, lengthOfFile/2, 0);
48     if(bSuccess) {
49         printf("shared: File half lock on %s released.\n", szFilename);
50     }
51     CloseHandle(hFileHandle);
52
53     ///exclusive lock
54     HANDLE hFile = CreateFile(szFile, // lpFileName
55         GENERIC_READ | GENERIC_WRITE, // dwDesiredAccess
56         FILE_SHARE_WRITE, // dwShareMode
57         0, // lpSecurityAttributes
58         CREATE_ALWAYS, // dwCreationDisposition
59         0, // dwFlagsAndAttributes
60         0
61     );
62     OVERLAPPED overlapvar;
63     overlapvar.Offset = 0;
64     overlapvar.OffsetHigh = 0;
65
66     ///Buten tsociloh
67     bSuccess=LockFileEx(hFile, 0, 0, sizeof(szBuffer), 0, &overlapvar);
68     if(bSuccess) {
69         printf("\n\nexclusive: File lock on %s secured.\n", szFile);
70     }
71     bSuccess = UnlockFile(hFile, 0, 0, sizeof(szBuffer), 0);
72     if(bSuccess) {
73         printf("exclusive: File lock on %s released.\n", szFile);
74     }
75
76     ///hagas tsociloh
77     bSuccess=LockFileEx(hFile, 0, 0, sizeof(szBuffer)/2, 0, &overlapvar);
78     if(bSuccess) {
79         printf("exclusive: File half lock on %s secured.\n", szFile);
80     }
81     bSuccess = UnlockFile(hFile, 0, 0, sizeof(szBuffer)/2, 0);
82     if(bSuccess) {
83         printf("exclusive: File half lock on %s released.\n", szFile);
84     }
85 }

```

```
85     CloseHandle(hFile);  
86     return 0;  
87 }  
88  
89
```

```
shared: File lock on testing.txt secured.  
shared: File lock on testing.txt released.  
shared: File half lock on testing.txt secured.  
shared: File half lock on testing.txt released.
```

```
exclusive: File lock on test1.txt secured.  
exclusive: File lock on test1.txt released.  
exclusive: File half lock on test1.txt secured.  
exclusive: File half lock on test1.txt released.
```

```
Process returned 0 (0x0)   execution time : 0.325 s  
Press any key to continue.
```