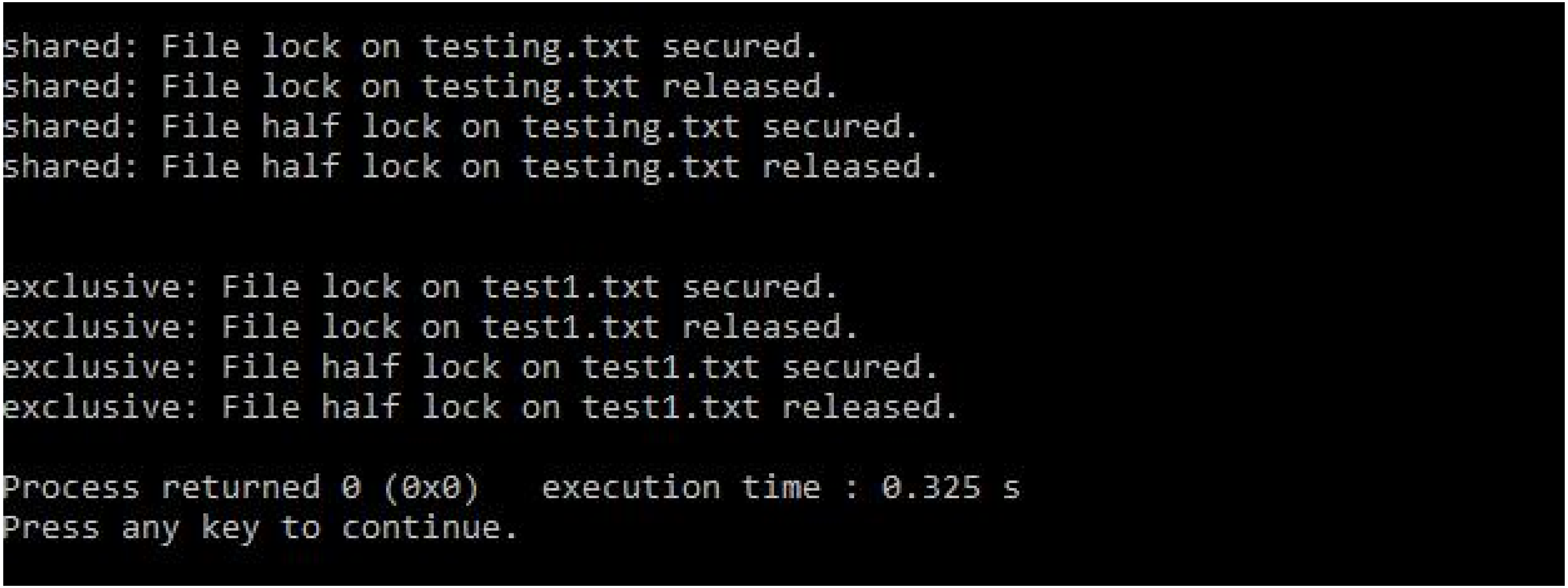
```c
#include <Windows.h>
#include <stdio.h>

int main(void){
    //shared lock
    HANDLE hFileHandle;
    INT EXCLUSIVE = 0, SHARED = 1;
    BOOL bSuccess;
    char *szFilename = "testing.txt";
    char *szFile = "test1.txt";
    char szBuffer[245];

    ///file uusgeh
    hFileHandle = CreateFile(szFilename, GENERIC_READ,
        FILE_SHARE_READ, 0, CREATE_ALWAYS, 0, 0);
    if(hFileHandle == INVALID_HANDLE_VALUE){
        printf("Shine file uusgej chdsangui.\n");
        exit(EXIT_FAILURE);
    }
    FILE *fptr = fopen("testing.txt","rb");
    if(fptr == NULL)
    {
          printf("Error!");
          exit(1);
    }
    fprintf(fptr,"%d", 1234);
    fseek(fptr, 0L, SEEK_END);
    int lengthOfFile = ftell(fptr);
    fclose(fptr);

    ///bufferiin urtaar file lock hiine. Buten lock
    bSuccess = LockFile(hFileHandle, 0, 0, lengthOfFile,0);
    if(bSuccess){
        printf("\n\nshared: File lock on %s secured.\n", szFilename);
    }
    bSuccess = UnlockFile(hFileHandle, 0, 0, lengthOfFile, 0);
    if(bSuccess){
        printf("shared: File lock on %s released.\n", szFilename);
    }

    /// hagas lock
    bSuccess = LockFile(hFileHandle, 0, 0, lengthOfFile/2,0);
    if(bSuccess){
        printf("shared: File half lock on %s secured.\n", szFilename);
    }

    bSuccess = UnlockFile(hFileHandle, 0, 0, lengthOfFile/2, 0);
    if(bSuccess){
        printf("shared: File half lock on %s released.\n", szFilename);
    }
    CloseHandle(hFileHandle);

    ///exclusive lock
    HANDLE hFile = CreateFile(szFile,    // lpFileName
        GENERIC_READ | GENERIC_WRITE,    // dwDesiredAccess
        FILE_SHARE_WRITE,               // dwShareMode
        0,                              // lpSecurityAttributes
        CREATE_ALWAYS,                  // dwCreationDisposition
        0,                              // dwFlagsAndAttributes
        0
    );
    OVERLAPPED overlapvar;
    overlapvar.Offset = 0;
    overlapvar.OffsetHigh = 0;

    /// Buten tsoojloh
    bSuccess=LockFileEx(hFile, 0, 0, sizeof(szBuffer),0,&overlapvar);
    if(bSuccess){
        printf("\n\nexclusive: File lock on %s secured.\n", szFile);
    }
    bSuccess = UnlockFile(hFile, 0, 0, sizeof(szBuffer), 0);
    if(bSuccess){
        printf("exclusive: File lock on %s released.\n", szFile);
    }

    /// hagas tsoojloh
    bSuccess=LockFileEx(hFile, 0, 0, sizeof(szBuffer)/2,0,&overlapvar);
    if(bSuccess){
        printf("exclusive: File half lock on %s secured.\n", szFile);
    }
    bSuccess = UnlockFile(hFile, 0, 0, sizeof(szBuffer)/2, 0);
    if(bSuccess){
        printf("exclusive: File half lock on %s released.\n", szFile);
    }
    CloseHandle(hFile);
    return 0;
}


```



```
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.
```