

מבוא למערכות הפעלה – תרגיל 2

מרצה: דר' כרמי מרימוביץ

שימו לב: תרגיל זה דורש לקרוא תיעוד של Win32.

1. בתרגיל זה עליכם לממש מערכת מעטפות (Folder, Directory=). מעטפה אמיתית אחת של המערכת תשמש כדיסק.
2. קובץ אחד (נאמר "0") יכיל את מבנה המעטפות. שימו לב, המבנה צריך להיות עץ ללא מגבלת רמות, וכמובן שחיפוש בו צריך להיות **יעיל!**
3. לכל קובץ צריך להיות שם (כמובן!) ומצביע למקום הקובץ. פיזית, כל קובץ יקבל כשם אמיתי מספר רץ (1, 2, 3) ושם זה יהיה המצביע.
4. עליכם לספק פונקציות לביצוע: Open, Create, Close, Read, Write, Delete, Rename.
5. יש לספק פונקציית איתחול כלשהיא של מעטפה אמיתית (סוג של Format).
6. לגבי Open, Close, Create, Read, Write, עיטפו את אחת ממערכות ה-I/O של C. לדוגמא fopen, fclose, fread, fwrite.
7. אין צורך להשתמש שוב במיספר של קובץ שנמחק. (אבל אין איסור).
8. במידה וב-"0" נכמר המקום יש להגדילו.
9. הבעיה **הממשית** בתרגיל היא ניהול הקובץ "0". (ועל זה דיברנו **שעות**).
10. כמובן עליכם לספק תוכנית שתדגים את כל הפונקציות הנ"ל.
11. פתיחת קובץ (ויצירתו אם אינו קיים) עבור מפוי ב-Win32.

```
#include <windows.h>
```

```
hFile = CreateFile(
    "Filename",
    GENERIC_READ|GENERIC_WRITE,
    0,           // no sharing
    NULL,       // security
    OPEN_ALWAYS,
    FILE_ATTRIBUTE_NORMAL,
    NULL);
```

12. סגירת קובץ ב-Win32.

```
#include <windows.h>
```

```
CloseHandle(hMapFile);
```

13. יצירת מפוי לקובץ ב-Win32.

```
hMapFile = CreateFileMapping(  
    hFile,          // file handle  
    NULL,          // security  
    PAGE_READWRITE,  
    0,             // all file  
    0,  
    NULL);  
buf = MapViewOfFile(  
    hMapFile,       // mapping handle  
    FILE_MAP_ALL_ACCESS,  
    0,             // all file  
    0,  
    0);
```

14. סיום המפוי ב-Win32.

```
#include <windows.h>  
UnmapViewOfFile(buf);  
CloseHandle(hFile);
```

שימו לב, כדאי מאוד (מאוד) לקרוא את התיעוד של הפונקציות הנ"ל ולא להשתמש בהן באופן לא מושכל.

אין לעבוד בקבוצות

התיעוד משמעותי לציון.

שפות רלוונטיות: C/C++ ללא MFP/ATL וכיו"ב.

את התרגילים יש להגיש לתיבת התרגילים של הקורס.