

16/9/2007

### מבוא למערכות הפעלה – תרגיל 3

מרצה: ד"ר כרמי מרימוביץ.

1. בתרגיל זה עליכם לממש מערכת מעטפות (Folder, Directory). מעטפה אמיתית אחת של המערכת בה אתם משתמשים תשמש כדיסק.
2. קובץ אחד (נאמר ששמו "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: Create, Open, Close, Read, Write, Delete, Rename, Format.
5. ב-Format יש למחוק את כל הקבצים וליצור את הקובץ "0" מאותחל.
6. לגבי Read, Open, Close ו-Write: תעטפו אחת ממערכות ה-I/O של C. לדוגמא .fopen/fclose/fread/fwrite
7. כאשר מוחקים קובץ אין להשתמש במספר שלו שוב.
8. במידה ונגמר המקום ב-"0", יש להגדיל את "0".
9. הבעיה **הממשית** בתרגיל היא ניהול הקובץ "0". שימוש בקובץ ממופה זכרון (Memory mapped file) יעשה את חייכם קלים מבחינה זו.
10. כמובן עליכם לספק תוכנית שתפעיל את כל הפונקציות הללו על-ידי משתמש.
11. כרגיל, יש לכתוב את התוכניות פעם ב-Win32 ופעם ב-Posix.
12. פתיחת קובץ (ויצירתו אם אינו קיים) עבור מפוי ב-Win32.

```
#include <windows.h>

hMapFile = CreateFile(
    "File name",
    GENERIC_READ | GENERIC_WRITE,
    0,           // No sharing
    NULL,       // Security
    OPEN_ALWAYS,
    FILE_ATTRIBUTE_NORMAL,
    NULL
```

```
);
```

13. סגירת הקובץ ב-Win32.

```
#include <windows.h>  
CloseHandle(hMapFile);
```

14. יצירת מפוי לקובץ ב-Win32.

```
#include <windows.h>  
  
hMapFile = CreateFileMapping(  
    hFile,          // file handle  
    NULL,           // security  
    PAGE_READWRITE,  
    0,              // All file  
    0,  
    NULL            // Shared object name  
);
```

```
buf = MapViewOfFile(  
    hMapFile,       // Mapping handle  
    FILE_MAP_ALL_ACCESS,  
    0,              // All file  
    0,  
    0  
);
```

15. סיום המיפוי ב-Win32.

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

16. יצירת מפוי Posix.

```
#include <sys/mman.h>
```

```

buf = mmap(
    0,          // Hint for start address
    Length,
    PROT_READ|PROT_WRITE,
    MAP_SHARED, // Update file on writing
    fd,         // File descriptor
    0           // Offset
);

```

17. סיום מפוי ב-Posix.

```

#include <sys/mman.h>
munmap(buf, len);

```

18. ב-Posix יש להשתמש ב-open/create/close כדי לגשת לקבצים.

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

**אין לעבוד בקבוצות.**

התיעוד משמעותי לציון.

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