

מעבדה 13 - קבצים

הוספת שמירה לקובץ לתוכנית הספרייה

- **בעיה:** הנתונים שהמשתמש מקליד בכל הרצה של תכנית הספרייה שכתבנו הולכים לאיבוד בסופה. בכל הרצה של התכנית מתחילים מאפס.
- כיצד נשמור את הנתונים בין ריצות עוקבות של התכנית?
- **פתרון:** נשמור את הנתונים לקובץ, ובהרצת התכנית נקרא אותם מתוכו.

הוספת פונקציות שמירה Save לתוכנית הספרייה

- נוסף למודול book פונקציה לשמירת רשומת ספר בודד לקובץ בפורמט טקסט:

```
void book_save_to_file(FILE *fp, BookPtr pbook)
{
    fprintf(fp, "\n%s\n%s\n%s\n%d\n%f\n%d",
            pbook->name,
            pbook->writer,
            pbook->publisher,
            pbook->year,
            pbook->cost,
            pbook->ID);
}
```

- נתוני הספר נכתבים לקובץ ע"י fprintf נתון בשורה.

הוספת פונקציות שמירה Save לתוכנית הספרייה

■ למודול bookarr נוסיף פונקציה לשמירת כל מערך הספרים לקובץ:

```
void array_save_to_file(Array *parray, char *file_name)
{
    FILE* fp;
    /* open file for writing in text mode */
    if( (fp = fopen(file_name, "wt")) == NULL)
    {
        puts("couldn't open file!");
        return;
    }
    /* write the book list to the file */
    ....
    fclose(fp);
}
```

פונקציות טעינה מהקובץ

- מאחר ששמרנו את נתוני התכנית לקובץ, יש להגדיר פונקציה שקוראת אותם מהקובץ למשתנים בזיכרון.
- לפני כן, נגדיר פונקצית עזר לקריאת שורה מקובץ:

```
/* a utility function - read a line from an input file and return it */
char *getline(FILE *fp)
{
    String temp;

    fgets(temp, MAX_STR_LEN, fp);
    temp[strlen(temp)-1] = '\0'; /* instead of the '\n' inserted by fgets */
    return strdup(temp);
}
```

- הסבר: הפונקציה קוראת שורת טקסט מהקלט ע"י fgets ומחזירה שיכפול שלה - ע"י strdup - לפונקציה הקוראת.
- לפני ההחזרה, הפונקציה מתקנת בעיה שנגרמת ע"י הפונקציה fgets(): השורה שנקראה מהקלט מכילה גם את תו השורה החדשה '\n' כתו אחרון במחרוזת (לפני תו סיום המחרוזת '\0').
- כדי למחוק את התו האחרון '\n' מציבים את תו סיום המחרוזת במקומו.

טעינת נתונים מקובץ

- הפונקציה הקוראת רשומת ספר מהקובץ מוגדרת כך במודול book :

```
void book_read_from_file(FILE *fp, BookPtr pbook)
{
    String temp;
    pbook->name = getline(fp);
    pbook->writer = getline(fp);
    pbook->publisher = getline(fp);
    fscanf(fp, "%d", &pbook->year);
    fscanf(fp, "%f", &pbook->cost);
    fscanf(fp, "%d", &pbook->ID);
    fgets(temp, MAX_STR_LEN, fp); /* to read up to the end of line */
}
```

```
void array_load_from_file(Array *parray, char *file_name)
{
    FILE* fp;
    String temp;
    /* open file for reading in text mode */
    if( (fp = fopen(file_name, "rt")) == NULL)
        return;
    /* read the book array from the file */
    ...
    fgets(temp, MAX_STR_LEN, fp); /* read first heading line */
    while(!feof(fp))
    {
        ...
        book_read_from_file(fp, p->pbook);
        plist->count++;
    }
    p->next = NULL ; /* last element points to NULL */
    fclose(fp);
}
```

- במודול bookarray מוגדרת הפונקציה הטוענת את כל רשומות הספרים מהקובץ לרשימה המקושרת :

הפונקציה הראשית main

- הקריאה לפונקצית השמירה לקובץ מבוצעת בתכנית הראשית כתגובה לבחירת המשתמש באופציה s - שמירה לקובץ - שהוספה לתפריט:

```
void menu()
{
    printf("\n\n\n\n\n");
    puts("Book program main menu");
    puts("-----");

    ...
    puts("s. Save books records to file");
    puts("x. Exit");
    puts("\n\n\n\n\n\n\n\n");
}
```

- הקריאה לפונקצית הטעינה לעומת זאת, נקראת **תמיד** בתחילת התכנית בכדי לטעון את הנתונים הקיימים:

```
array_init(&array);

- /* restore old data from file */
array_load_from_file(&array, DATA_FILE_NAME);
```


שמירה ביציאה מהתכנית

- המשתמש יכול כעת לבצע שמירה יזומה לתכנית, בכל רגע שירצה. אולם מה אם הוא ישכח ויבקש לצאת מהתכנית לפני ששמר את השינויים האחרונים שביצע?
- כאשר המשתמש מבקש לסיים את התכנית, נהוג לבדוק אם הוא ביצע שינויים במבני הנתונים מאז השמירה האחרונה ואם כן, לשאול אותו האם ברצונו לבצע שמירה. עפ"י החלטתו מבוצעת השמירה.
- **שאלה:** כיצד נדע בבקשת היציאה אם בוצע שינוי בתכנית מאז השמירה האחרונה?
- **תשובה:** נחזיק משתנה בוליאני - בעל ערך "אמת" או "שקר" - שיציין בכל רגע נתון האם נתוני התכנית שונו מאז השמירה האחרונה או לא.
- ערכו של הנתון יהיה בתחילת התכנית "שקר", ולאחר כל ביצוע שינוי - קרי הוספת ספר או הסרת ספר - הוא ישונה ל- "אמת".
- לאחר שמירת הנתונים לקובץ שוב יוצב למשתנה ערך "שקר".

שמירה ביציאה מהתכנית

- נגדיר משתנה בוליאני בטיפוס המערך שיציין האם נתוני הרשימה שונו מאז השמירה האחרונה לקובץ:

```
typedef struct
{
    BookPtr *array; /* pointer to the book array */
    int size;
    int count; /* number of elements in the array */
    Boolean changed_fl; /* array change flag */
    int count;
} Array;
```

- ערכו של המשתנה מאותחל ל- FALSE בפונקציה list_init() בתחילת התוכנית מצב נתוני הספריה הוא "לא שונו".
- המשתנה יעודכן בפונקציות **המשנות** את הנתונים ברשימה.
- נצטרך לכתוב פונקציה שתבדוק האם יש לשמור את הרשימה לקובץ

```
void array_check_for_save(List *plist, char *file_name);
```