



מבוא לתכנות בשפת C

קבצים



קבצים

- לשמור נתונים לטווח ארוך
- לשמור כמות גדולה של נתונים
- לשתף נתונים בין תכניות שונות



אופן השימוש

- פתיחה
- קריאה וכתובה
- סגירה



פתיחה של קובץ

```
FILE *fopen ( const char *name , const char *mode );
```

- בזמן הפתיחה יש לציין את הדברים הבאים:
 - name – שם הקובץ
 - mode – אופן השימוש הנדרש בקובץ
 - “r” – לקריאה בלבד
 - שגיאה אם הקובץ לא קיים
 - “w” – לכתיבה בלבד
 - אם הקובץ לא קיים נוצר חדש, ואם הוא קיים הוא נדרס
 - “a” – להוספה
 - אפשר גם “r+”, “w+” ו- “a+”
- אם הפתיחה הצליחה הפונקציה מחזירה מצביע ל- FILE, אחרת NULL



סגירה של קובץ

```
int fclose ( FILE *fp );
```

- לאחר השימוש יש לסגור הקובץ



קריאה וכתובה מקובץ

```
int fgetc( FILE *fp );  
int fputc( int c, FILE *fp );  
char *fgets( char *s, int n, FILE *fp );  
int fputs( const char *s, FILE *fp );  
int fscanf ( FILE *fp , const char *format , ... );  
int fprintf ( FILE *fp , const char *format , ... );
```

- נראה את אופן השימוש המדויק בהמשך



דוגמה נרחבת

```
#include <stdio.h>
#include <stdlib.h>

#define INFN "input.txt"
#define OUTFN "output.txt"

#define MAX_LINE_SIZE 80

#define EOLN '\\r'
```



העתקת קובץ תו אחרי תו

```
void copyFileCharByChar() {  
    FILE *in, *out;  
    char c;  
    if ((in = fopen(INFN, "r")) == NULL) {  
        perror(INFN);  
        exit(1);  
    }  
    if ((out = fopen(OUTFN, "w")) == NULL) {  
        perror(OUTFN);  
        exit(1);  
    }  
    while ((c = fgetc(in)) != EOF)  
        fputc(c, out);  
    fclose(in);  
    fclose(out);  
}
```




העתקת קובץ שורה אחר שורה

```
void copyFileLineByLine() {
    FILE *in, *out;
    char line[MAX_LINE_SIZE + 1];
    if ((in = fopen(INFN, "r")) == NULL) {
        perror(INFN);
        exit(1);
    }
    if ((out = fopen(OUTFN, "w")) == NULL) {
        perror(OUTFN);
        exit(1);
    }
    // while((fgets(line, MAX_LINE_SIZE+1, in)) != NULL)
    while ((fgets(line, MAX_LINE_SIZE, in)))
        fputs(line, out);
    fclose(in);
    fclose(out);
}
```

חילוץ מלים מקובץ

```
void listOfWords() {
    FILE *in, *out;
    char line[MAX_LINE_SIZE + 1];
    if ((in = fopen(INFN, "r")) == NULL) {
        perror(INFN);
        exit(1);
    }
    if ((out = fopen(OUTFN, "w")) == NULL) {
        perror(OUTFN);
        exit(1);
    }
    while ((fgets(line, MAX_LINE_SIZE + 1, in))) {
        char *p = line;
        while (*p != EOLN) {
            // skip space
            while (*p == ' ' || *p == '\t')
                ++p;
            // write a word
            if (*p != EOLN) {
                while (*p != EOLN && *p != ' ' && *p != '\t')
                    fputc(*p++, out);
                fputc('\n', out);
            }
        }
    }
    fclose(in);
    fclose(out);
}
```



חילוץ מלים מקובץ

```
void listOfWords1() {  
    FILE *in, *out;  
    char line[MAX_LINE_SIZE + 1];  
    if ((in = fopen(INFN, "r")) == NULL) {  
        perror(INFN);  
        exit(1);  
    }  
    if ((out = fopen(OUTFN, "w")) == NULL) {  
        perror(OUTFN);  
        exit(1);  
    }  
    while (fscanf(in, "%s", line) != EOF)  
        fprintf(out, "%s\n", line);  
    fclose(in);  
    fclose(out);  
}
```

הדפסה יפה (דילוג על רווחים מיותרים)

```
void prityPrint() {
    FILE *in, *out;
    char line[MAX_LINE_SIZE + 1];
    if ((in = fopen(INFN, "r")) == NULL) {
        perror(INFN); exit(1);
    }
    if ((out = fopen(OUTFN, "w")) == NULL) {
        perror(OUTFN); exit(1);
    }
    while ((fgets(line, MAX_LINE_SIZE + 1, in))) {
        char *org = line, *rslt = line;
        int firstWord = 1;
        while (*org != EOLN) {
            // skip space
            while (*org == ' ' || *org == '\t')
                ++org;
            // add one space between words
            if (!firstWord && *org != EOLN && *org != ' ' && *org != '\t')
                *rslt++ = ' ';
            // copy a word
            while (*org != EOLN && *org != ' ' && *org != '\t')
                *rslt++ = *org++;
            firstWord = 0;
        }
        // add a new line and end of string
        *rslt++ = '\n';
        *rslt = '\0';
        // write it down
        fputs(line, out);
    }
    fclose(in);
    fclose(out);
}
```

קובץ רשומות בפורמט קשיח

```
void fixRecordFile() {
    FILE *in, *out;
    char name[17], addr[17];
    int id, n;

    if ((in = fopen(INFN, "r")) == NULL) {
        perror(INFN);
        exit(1);
    }
    if ((out = fopen(OUTFN, "w")) == NULL) {
        perror(OUTFN);
        exit(1);
    }

    while (fscanf(in, "%5d%16[^$]%16[^$]%2d%c", &id, name, addr, &n) != EOF)
        fprintf(out, "%5d%16s%16s%2d\n", id, name, addr, n);

    fclose(in);
    fclose(out);
}
```

| | | | | | | |
|-------|--|------------------|--|------------------|--|----|
| 12345 | | Haiim Ben Yehuda | | Harakefet st. BS | | 2 |
| 54321 | | Bella | | Narkis 14, TL | | 3 |
| 125 | | Tami Seri | | Kibutz | | 12 |