



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κεντρικής Μακεδονίας - Σέρρες
Τμήμα Μηχανικών Πληροφορικής

Προγραμματισμός II (Θ)

Δρ. Δημήτρης Βαρσάμης
Επίκουρος Καθηγητής

Μάρτιος 2017

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ II (Θ)

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Αντικείμενο της παρούσης ενότητας είναι τα αρχεία δεδομένων. Θα αναφερθούμε

- για το πώς τα χρησιμοποιούμε στον προγραμματισμό
- πώς τα δημιουργούμε
- πώς τα διαβάζουμε
- πώς τα ενημερώνουμε κ.α.

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Αρχεία (Files)

- Η αποθήκευση δεδομένων σε μεταβλητές και πίνακες χαρακτηρίζεται ως μια προσωρινή λύση.
- Όλα αυτά τα δεδομένα που έχουν αποθηκευτεί σε μεταβλητές ή πίνακες ή σε δομές, χάνονται έπειτα από τον τερματισμό του προγράμματος.
- Για να αποφύγουμε επομένως την απώλεια δεδομένων, χρησιμοποιούμε για την αποθήκευση τους τα **αρχεία**.
- Η απώλεια των δεδομένων, οφείλεται στο ότι το πρόγραμμα έχει την δυνατότητα να αποθηκεύει μόνο στην προσωρινή μνήμη RAM.

Αρχεία (Files)

- Τα αρχεία όμως λύνουν το πρόβλημα της προσωρινής αποθήκευσης δεδομένων διότι, μας παρέχουν την δυνατότητα τα δεδομένα να αποθηκεύονται κάπου μόνιμα, όπως για παράδειγμα στον σκληρό δίσκο ή σε κάποια άλλη περιφερειακή συσκευή.
- Σε κάθε γλώσσα προγραμματισμού υπάρχει μία βιβλιοθήκη η οποία μας επιτρέπει την πρόσβαση στα αρχεία, στην C αυτή η βιβλιοθήκη είναι η `stdio.h`.
- Η βιβλιοθήκη αυτή επικοινωνεί με το σύστημα διαχείρισης αρχείων (File System) του λειτουργικού συστήματος και πραγματοποιεί τις εκάστοτε λειτουργίες που ζητάει ο προγραμματιστής.

Αρχεία (Files)

Τα αρχεία χωρίζονται σε δύο κατηγορίες:

- **Δυαδικά (binary)**
- **Κειμένου (text)**

Τα **δυαδικά αρχεία**, είναι αρχεία των οποίων η μορφή τους είναι μία **ακολουθία από χαρακτήρες (bytes)** και τα δεδομένα αναπαρίστανται όπως ακριβώς και στην μνήμη.

Για παράδειγμα όταν εμείς γράφουμε την λέξη **file** στο σύστημα μεταφράζεται ως εξής:

01100110 01101001 01101100 01100101

Αρχεία (Files)

Όταν όμως αναφερόμαστε σε **αρχεία κειμένου** οι χαρακτήρες αποθηκεύονται **σειριακά** και διαχωρίζονται μεταξύ τους από ειδικούς χαρακτήρες.

Στα αρχεία κειμένου χρησιμοποιείται το αναγνωριστικό **EOF** το οποίο σημαίνει End Of File και υποδηλώνει το τέλος του αρχείου, σε αυτό το σημείο επομένως τερματίζεται και η ανάγνωση του αρχείου. Για παράδειγμα εάν είχαμε ένα αρχείο το οποίο είχε την εξής μορφή:

learning about files

αυτό στο αρχείο μας θα ήταν γραμμένο με τον εξής τρόπο:

`learning\tabout\tfilesEOF`

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Άνοιγμα επεξεργασίας αρχείου

- Για να ανοίξουμε ένα αρχείο χρησιμοποιούμε την συνάρτηση `fopen()`, η οποία μας παρέχει την δυνατότητα να δημιουργήσουμε το αρχείο, εφόσον αυτό δεν υπάρχει.
- Το πρωτότυπο της συνάρτησης είναι

```
FILE *fopen(char *filename, char *action);
```

- Με αυτήν την συνάρτηση, αρχικοποιούμε ένα δείκτη σε αρχείο ο οποίος είναι τύπου `FILE` και δέχεται δύο ορίσματα, το όνομα του αρχείου και τον τρόπο με τον οποίο επιθυμούμε να το χρησιμοποιήσουμε.
- Για παράδειγμα με την παρακάτω γραμμή κώδικα

```
FILE *fp = fopen("test.txt", "r");
```

δηλώνουμε ένα δείκτη αρχείου (`FILE`) `fp` στον οποίο εκχωρούμε το άνοιγμα του αρχείου `test.txt` με δυνατότητα ανάγνωσης `r`.

Άνοιγμα επεξεργασίας αρχείου

- Διαφορετικά, θα μπορούσαμε τα γράψουμε ξεχωριστά

```
FILE *fp;  
fp = fopen("test.txt", "r");
```

Άνοιγμα επεξεργασίας αρχείου

Οι βασικοί τρόποι με τους οποίους είναι δυνατόν να ανοιχτεί ένα **αρχείο εγγραφών** ακολουθούν παρακάτω

Ενέργεια	Περιγραφή
r	Άνοιγμα αρχείου για ανάγνωση.
w	Δημιουργία αρχείου για εγγραφή. Εάν το αρχείο υπάρχει ήδη τότε τα περιεχόμενα του διαγράφονται.
a	Προσάρτηση. Άνοιγμα ή δημιουργία αρχείου για εγγραφή στοιχείων στο τέλος του.
r+	Άνοιγμα αρχείου για ενημέρωση. Ανάγνωση και εγγραφή.
w+	Για εγγραφή στο αρχείο αλλά και δημιουργία του εάν δεν υπάρχει. Εάν το αρχείο υπάρχει ήδη τότε τα περιεχόμενα του διαγράφονται.
a+	Προσάρτηση. Ανοίγει ή δημιουργεί ένα αρχείο για ενημέρωση. Η εγγραφή στοιχείων γίνεται στο τέλος του.

Άνοιγμα επεξεργασίας αρχείου

Οι βασικοί τρόποι με τους οποίους είναι δυνατόν να ανοιχτεί ένα **δυναμικό αρχείο** ακολουθούν παρακάτω

Ενέργεια	Περιγραφή
rb	Άνοιγμα αρχείου για ανάγνωση.
wb	Δημιουργία αρχείου για εγγραφή. Εάν το αρχείο υπάρχει ήδη τότε τα περιεχόμενα του διαγράφονται.
ab	Προσάρτηση. Άνοιγμα ή δημιουργία αρχείου για εγγραφή στοιχείων στο τέλος του.
r+b	Άνοιγμα αρχείου για ενημέρωση. Ανάγνωση και εγγραφή.
w+b	Για εγγραφή στο αρχείο αλλά και δημιουργία του εάν δεν υπάρχει. Εάν το αρχείο υπάρχει ήδη τότε τα περιεχόμενα του διαγράφονται.
a+b	Προσάρτηση. Ανοίγει ή δημιουργεί ένα αρχείο για ενημέρωση. Η εγγραφή στοιχείων γίνεται στο τέλος του.

Άνοιγμα επεξεργασίας αρχείου - Παραδείγματα

- `fp = fopen("test.txt", "r");`

άνοιγμα του αρχείου κειμένου `test.txt` με δυνατότητα ανάγνωσης `r`.

- `fp = fopen("test.txt", "a+");`

άνοιγμα του αρχείου κειμένου `test.txt` με δυνατότητα προσάρτησης και ανάγνωσης `r`.

- `fp = fopen("test.dat", "wb");`

άνοιγμα του δυαδικού αρχείου `test.dat` με δυνατότητα εγγραφής `w`.

- `fp = fopen("d:\\prog2\\test.dat", "wb");`

Άνοιγμα επεξεργασίας αρχείου - Παραδείγματα

άνοιγμα του δυαδικού αρχείου `test.dat` με δυνατότητα εγγραφής `w` το οποίο βρίσκεται στην διαδρομή `d:\prog2\`.

```
fp = fopen("test.dat", "rb");  
if (fp==NULL) {  
    printf("Error: The file can not be opened\n");  
}
```

έλεγχος αν άνοιξε το δυαδικό αρχείο `test.dat` με δυνατότητα ανάγνωσης `r`.

¹Τα παραπάνω ισχύουν σε λειτουργικό σύστημα Windows

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Κλείσιμο επεξεργασίας αρχείου

- Το κλείσιμο κάποιου αρχείου που ανοίξαμε είναι υποχρεωτικό, εάν το παραβλέψουμε θα δημιουργηθούν λάθη και προβλήματα στο πρόγραμμά μας.
- Το κλείσιμο ενός αρχείου είναι μια απλή διαδικασία, επιτυγχάνεται με την χρήση της συνάρτησης `fclose()` που μας παρέχει η βιβλιοθήκη `stdio.h`.
- Η συνάρτηση `fclose()` όπως και η συνάρτηση `fopen()` χρησιμοποιούνται και στα δυαδικά και στα αρχεία κειμένου.
- Το πρωτότυπο της συνάρτησης είναι

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

και η σύνταξη της εντολής είναι η εξής:

```
|| fclose(fp);
```

Κλείσιμο επεξεργασίας αρχείου

η οποία δέχεται σαν όρισμα το δείκτη αρχείου που έχει συσχετιστεί με ένα ανοικτό αρχείο.

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Εγγραφή σε Αρχεία κειμένου

Οι συναρτήσεις που χρησιμοποιούνται για την εγγραφή δεδομένων σε αρχεία κειμένου είναι: `fputc()`, `fputs()`, `fprintf()`.

- Η συνάρτηση `fputc()` εγγράφει ένα χαρακτήρα σε ένα αρχείο
- Το πρωτότυπο της συνάρτησης είναι

```
int fputc(int ch, FILE *fp);
```

και η σύνταξη της εντολής είναι η εξής:

```
|| fputc('a', fp);
```

η οποία δέχεται σαν ορίσματα ένα χαρακτήρα το δείκτη αρχείου που έχει συσχετιστεί με ένα ανοικτό αρχείο. Όταν η εγγραφή είναι επιτυχής τότε επιστρέφει μη αρνητική τιμή, διαφορετικά επιστρέφει `EOF`.

- Για παράδειγμα το παρακάτω πρόγραμμα

Εγγραφή σε Αρχεία κειμένου

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

int main() {
    char str[] = "programmatismos II";
    int i;
    FILE *fp;
    fp = fopen("test1.txt", "w");
    if (fp==NULL) {
        printf("Error: fopen() failed\n");
        exit(EXIT_FAILURE);
    }
    for (i=0; str[i]!='\0'; i=i+1) {
        fputc(str[i], fp);
    }
    fclose(fp);
    return 0;
}
```


Εγγραφή σε Αρχεία κειμένου

|| }

εγγράφει το `str` ανά χαρακτήρα στο αρχείο `test1.txt`.

- Ανοίγουμε το αρχείο κειμένου μέσω του Notepad το οποίο περιέχει
programmatismos II

- Το μέγεθος του αρχείου είναι 18 Bytes

Εγγραφή σε Αρχεία κειμένου

- Η συνάρτηση `fputs()` εγγράφει ένα αλφαριθμητικό σε ένα αρχείο
- Το πρωτότυπο της συνάρτησης είναι

```
int fputs(const char *str, FILE *fp);
```

και η σύνταξη της εντολής είναι η εξής:

```
|| fputs(str, fp);
```

η οποία δέχεται σαν ορίσματα ένα αλφαριθμητικό και το δείκτη αρχείου που έχει συσχετιστεί με ένα ανοικτό αρχείο. Όταν η εγγραφή είναι επιτυχής τότε επιστρέφει μη αρνητική τιμή, διαφορετικά επιστρέφει `EOF`.

- Για παράδειγμα το παρακάτω πρόγραμμα

Εγγραφή σε Αρχεία κειμένου

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

int main() {
    char str[] = "programmatismos II";
    FILE *fp;
    fp = fopen("test2.txt", "w");
    if (fp==NULL) {
        printf("Error: fopen() failed\n");
        exit(EXIT_FAILURE);
    }
    fputs(str, fp);
    fclose(fp);
    return 0;
}
```

εγγράφει το αλφαριθμητικό `str` στο αρχείο `test2.txt`.

Εγγραφή σε Αρχεία κειμένου

- Ανοίγουμε το αρχείο κειμένου μέσω του Notepad το οποίο περιέχει
programmatismos II

- Το μέγεθος του αρχείου είναι 18 Bytes

Εγγραφή σε Αρχεία κειμένου

- Η συνάρτηση `fprintf()` εγγράφει μια μορφοποιημένη έκφραση σε ένα αρχείο. Η έκφραση μπορεί να αποτελείται από οποιοδήποτε τύπο δεδομένων.
- Το πρωτότυπο της συνάρτησης είναι

```
int fprintf(FILE *fp, const char *format, ...);
```

και η σύνταξη της εντολής είναι η εξής:

```
|| fprintf(fp, "%d %c\n", var1, var2);
```

η οποία δέχεται σαν ορίσματα το δείκτη αρχείου που έχει συσχετιστεί με ένα ανοικτό αρχείο, την επιθυμητή μορφοποίηση και την λίστα των μεταβλητών που χρησιμοποιούνται στην μορφοποίηση. Όταν η εγγραφή είναι επιτυχής τότε επιστρέφει μη αρνητική τιμή, διαφορετικά επιστρέφει `EOF`.

Εγγραφή σε Αρχεία κειμένου

- Προφανώς, η `fprintf()` συντάσσεται όπως η `printf()` με την μόνη διαφορά το πρώτο όρισμα της `fprintf()` στο οποίο δηλώνουμε σε ποια έξοδο θέλουμε να πάει η εγγραφή.
- οι παρακάτω εντολές είναι ισοδύναμες `fprintf(stdout, ...)`, `printf(...)`.
- Για παράδειγμα το παρακάτω πρόγραμμα

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

int main() {
    char str[] = "programmatismos II";
    int i;
    FILE *fp;
    fp = fopen("test3.txt", "w");
    if (fp==NULL) {
        printf("Error: fopen() failed\n");
    }
}
```

Εγγραφή σε Αρχεία κειμένου

```
        exit(EXIT_FAILURE);  
    }  
    for(i=0; str[i]!='\0'; i=i+1){  
        fprintf(fp, "%d: %c\n", i+1, str[i]);  
    }  
    fclose(fp);  
    return 0;  
}
```

εγγράφει κάθε χαρακτήρα του αλφαριθμητικού `str` με αρίθμηση στο αρχείο `test3.txt`.

Εγγραφή σε Αρχεία κειμένου

- Ανοίγουμε το αρχείο κειμένου μέσω του Notepad το οποίο περιέχει

1: p

2: r

3: o

4: g

5: r

6: a

7: m

8: m

9: a

10: t

11: i

12: s

13: m

14: o

15: s

Εγγραφή σε Αρχεία κειμένου

- Το μέγεθος του αρχείου είναι 117 Bytes

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Ανάγνωση σε Αρχεία κειμένου

Οι συναρτήσεις που χρησιμοποιούνται για την ανάγνωση δεδομένων από αρχεία κειμένου είναι: `fgetc()`, `fgets()`, `fscanf()`.

- Η συνάρτηση `fgetc()` διαβάζει ένα χαρακτήρα από ένα αρχείο
- Το πρωτότυπο της συνάρτησης είναι

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

και η σύνταξη της εντολής είναι η εξής:

```
|| ch=fgetc(fp);
```

η οποία δέχεται σαν όρισμα το δείκτη αρχείου που έχει συσχετιστεί με ένα ανοικτό αρχείο. Όταν η ανάγνωση είναι επιτυχής τότε επιστρέφει το χαρακτήρα που διαβάστηκε, διαφορετικά επιστρέφει `EOF`.

- Για παράδειγμα το παρακάτω πρόγραμμα

Ανάγνωση σε Αρχεία κειμένου

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

int main() {
    char str[18];
    int i;
    FILE *fp;
    fp = fopen("test1.txt", "r");
    if (fp==NULL) {
        printf("Error: fopen() failed\n");
        exit(EXIT_FAILURE);
    }
    for(i=0; i<18; i=i+1) {
        str[i]=fgetc(fp);
    }
    fclose(fp);
    printf("%s\n", str);
}
```

Ανάγνωση σε Αρχεία κειμένου

```
||      return 0;  
||  }
```

διαβάζει ανά χαρακτήρα από το αρχείο `test1.txt` και τα καταχωρεί στο `str`.

- θα εμφανιστεί στην κονσόλα

```
||programmatismos II
```

Ανάγνωση σε Αρχεία κειμένου

- Η συνάρτηση `fgets()` διαβάζει ένα αλφαριθμητικό από ένα αρχείο
- Το πρωτότυπο της συνάρτησης είναι

```
char *fgets(char *str, int size, FILE *fp);
```

και η σύνταξη της εντολής είναι η εξής:

```
|| fgets(str, sizeof(str), fp);
```

η οποία δέχεται σαν ορίσματα ένα δείκτη σε αλφαριθμητικό, το μέγεθος του και το δείκτη αρχείου που έχει συσχετιστεί με ένα ανοικτό αρχείο. Όταν η ανάγνωση είναι επιτυχής τότε επιστρέφει μη αρνητική τιμή, διαφορετικά επιστρέφει `NULL`.

- Για παράδειγμα το παρακάτω πρόγραμμα

Ανάγνωση σε Αρχεία κειμένου

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

int main() {
    char str[18];
    FILE *fp;
    fp = fopen("test2.txt", "r");
    if (fp==NULL) {
        printf("Error: fopen() failed\n");
        exit(EXIT_FAILURE);
    }
    fgets(str, sizeof(str), fp);
    fclose(fp);
    printf("%s\n", str);
    return 0;
}
```

Ανάγνωση σε Αρχεία κειμένου

διαβάζει από το το αρχείο `test2.txt` και το καταχωρεί στο αλφαριθμητικό `str`.

- θα εμφανιστεί στην κονσόλα

```
||programmatismos I
```

- Επειδή το αλφαριθμητικό `str` έχει 18 θέσεις κράτα την τελευταία για τον χαρακτήρα `\0`. Οπότε για να διαβάσει όλο το αλφαριθμητικό πρέπει να γράψουμε

`fgets(str, sizeof(str)+1, fp);`, το οποίο θα εμφανίσει στην κονσόλα

```
||programmatismos II
```


Ανάγνωση σε Αρχεία κειμένου

- Η συνάρτηση `fscanf()` διαβάζει μια μορφοποιημένη έκφραση σε ένα αρχείο. Η έκφραση μπορεί να αποτελείται από οποιοδήποτε τύπο δεδομένων.
- Το πρωτότυπο της συνάρτησης είναι

```
int fscanf(FILE *fp, const char *format, ...);
```

και η σύνταξη της εντολής είναι η εξής:

```
|| n=fscanf(fp, "%d", &var1);
```

η οποία δέχεται σαν ορίσματα το δείκτη αρχείου που έχει συσχετιστεί με ένα ανοικτό αρχείο, την επιθυμητή μορφοποίηση και την λίστα των μεταβλητών που χρησιμοποιούνται στην μορφοποίηση στις οποίες θα γίνει η εκχώρηση τιμών. Όταν η ανάγνωση είναι επιτυχής τότε επιστρέφει το πλήθος των στοιχείων που διάβασε, διαφορετικά επιστρέφει `EOF`.

Ανάγνωση σε Αρχεία κειμένου

- Προφανώς, η `fscanf()` συντάσσεται όπως η `scanf()` με την μόνη διαφορά το πρώτο όρισμα της `fscanf()` στο οποίο δηλώνουμε από ποια είσοδο θέλουμε να γίνει η ανάγνωση.
- οι παρακάτω εντολές είναι ισοδύναμες `fscanf(stdout, ...)`, `scanf(...)`.
- Για παράδειγμα το παρακάτω πρόγραμμα

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

int main() {
    char str[18];
    int i, temp;
    FILE *fp;
    fp = fopen("test3.txt", "r");
    if (fp==NULL) {
        printf("Error: fopen() failed\n");
    }
}
```

Ανάγνωση σε Αρχεία κειμένου

```
        exit(EXIT_FAILURE);  
    }  
    for(i=0; i<18; i=i+1) {  
        fscanf(fp, "%d: %c", &temp, &str[i]);  
    }  
    fclose(fp);  
    printf("%s\n", str);  
    return 0;  
}
```

διαβάζει από το αρχείο `test3.txt` τους χαρακτήρες και τους καταχωρεί στο αλφαριθμητικό `str` .

Ανάγνωση σε Αρχεία κειμένου

- Ανοίγουμε το αρχείο κειμένου μέσω του Notepad το οποίο περιέχει

1: p

2: r

3: o

4: g

5: r

6: a

7: m

8: m

9: a

10: t

11: i

12: s

13: m

14: o

15: s

Ανάγνωση σε Αρχεία κειμένου

- θα εμφανιστεί στην κονσόλα

```
|| programmatismos1II
```

- Αυτό που μας επιστρέφει το πρόγραμμα δεν είναι το επιθυμητό.
Ένα από τα προβλήματα που αντιμετωπίζουμε σε αρχεία κειμένου.

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Προσπέλαση σε Αρχεία κειμένου

- Η συνάρτηση `fseek()` χρησιμοποιείται για τυχαία προσπέλαση του αρχείου μετακινώντας τον δείκτη του αρχείου σε συγκεκριμένη θέση.
- Το πρωτότυπο της συνάρτησης είναι

```
int fseek(FILE *fp, long int offset, int origin);
```

και η σύνταξη της εντολής είναι η εξής:

```
|| fseek(fp, 0, SEEK_END);
```

η οποία δέχεται σαν ορίσματα το δείκτη αρχείου που έχει συσχετιστεί με ένα ανοικτό αρχείο, την επιθυμητή μετακίνηση κατά `offset` οκτάδες από το σημείο `origin` το οποίο είναι το τρίτο όρισμα. Η τιμή του ορίσματος `origin` πρέπει να είναι

- ▶ `SEEK_SET` η μετατόπιση του δείκτη ξεκινά από την αρχή του αρχείου.

Προσπέλαση σε Αρχεία κειμένου

- ▶ `SEEK_CUR` η μετατόπιση του δείκτη ξεκινά από την τρέχουσα θέση του δείκτη.
- ▶ `SEEK_END` η μετατόπιση του δείκτη ξεκινά από το τέλος του αρχείου.

Όταν η εκτέλεση είναι επιτυχής τότε επιστρέφει μηδέν, διαφορετικά μια μη μηδενική τιμή.

- Για παράδειγμα το παρακάτω πρόγραμμα

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

int main() {
    char ch1, ch2, ch3;
    FILE *fp;
    fp = fopen("test1.txt", "r");
    if (fp==NULL) {
        printf("Error: fopen() failed\n");
    }
}
```


Προσπέλαση σε Αρχεία κειμένου

```
        exit(EXIT_FAILURE);  
    }  
    fseek(fp, 10, SEEK_SET);  
    ch1=fgetc(fp);  
    ch2=fgetc(fp);  
    ch3=fgetc(fp);  
    printf("%c %c %c\n", ch1, ch2, ch3);  
    fclose(fp);  
    return 0;  
}
```

διαβάζει χαρακτήρες από συγκεκριμένες θέσεις του αρχείου `test1.txt`.

Προσπέλαση σε Αρχεία κειμένου

- Ανοίγουμε το αρχείο κειμένου μέσω του Notepad το οποίο περιέχει
programmatismos II

θα εμφανιστεί στην κονσόλα

```
|| i s m
```

Προσπέλαση σε Αρχεία κειμένου

- Ανοίγουμε το αρχείο κειμένου `test3.txt`. μέσω του Notepad το οποίο περιέχει

```
1: p  
2: r  
3: o  
4: g  
5: r  
6: a  
7: m  
8: m  
9: a  
10: t  
11: i  
12: s  
13: m  
14: o
```

Προσπέλαση σε Αρχεία κειμένου

θα εμφανιστεί στην κονσόλα

```
|| 3 :
```

Προσπέλαση σε Αρχεία κειμένου

- Η συνάρτηση `ftell()` μας επιστρέφει τη θέση του δείκτη του αρχείου.
- Το πρωτότυπο της συνάρτησης είναι

```
long int ftell(FILE *fp);
```

και η σύνταξη της εντολής είναι η εξής:

```
|| pos=ftell(fp);
```

η οποία δέχεται σαν ορίσματα το δείκτη αρχείου που έχει συσχετιστεί με ένα ανοικτό αρχείο και επιστρέφει την θέση.

- Για παράδειγμα το παρακάτω πρόγραμμα

Προσπέλαση σε Αρχεία κειμένου

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

int main() {
    long int pos;
    char str1[20];
    FILE *fp;
    fp = fopen("test4.txt", "r");
    if (fp==NULL) {
        printf("Error: fopen() failed\n");
        exit(EXIT_FAILURE);
    }
    while (fgetc(fp) != '\n');
    pos = ftell(fp);
    printf("POS= %ld\n", pos);
    fseek(fp, pos, SEEK_SET);
    fgets(str1, sizeof(str1), fp);
}
```

Προσπέλαση σε Αρχεία κειμένου

```
printf("%s\n", str1);  
fclose(fp);  
return 0;  
}
```

διαβάζει χαρακτήρες από συγκεκριμένες θέσεις του αρχείου `test4.txt`.

- Ανοίγουμε το αρχείο κειμένου μέσω του Notepad το οποίο περιέχει

```
programmatismos I  
programmatismos II
```

θα εμφανιστεί στην κονσόλα

Προσπέλαση σε Αρχεία κειμένου

```
|| POS= 19  
|| programmatismos II
```


Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Εγγραφή σε Δυναμικά αρχεία

- Η συνάρτηση `fwrite()` εγγράφει δεδομένα σε ένα δυναμικό αρχείο
- Το πρωτότυπο της συνάρτησης είναι

```
int fwrite(const void *buf, size_t size, size_t  
count, FILE *fp);
```

και η σύνταξη της εντολής είναι η εξής:

```
|| fwrite(arr, sizeof(int), 50, fp);
```

ή

```
|| fwrite(&a, sizeof(int), 1, fp);
```

η οποία δέχεται σαν ορίσματα

- ▶ την διεύθυνση μνήμης της μεταβλητής από την οποία θα εγγραφούν δεδομένα στο αρχείο²,

Εγγραφή σε Δυναμικά αρχεία

- ▶ το μέγεθος σε bytes για κάθε στοιχείου που θα εγγραφεί,
- ▶ τον αριθμό στοιχείων, το καθένα με μέγεθος bytes και
- ▶ το δείκτη αρχείου που έχει συσχετιστεί με ένα ανοικτό αρχείο.

Όταν η εγγραφή είναι επιτυχής τότε επιστρέφει το πλήθος των εγγραφών.

- Για παράδειγμα το παρακάτω πρόγραμμα

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

int main() {
    char str[] = "programmatismos II";
    int i, res;
    FILE *fp;
    fp = fopen("test1.dat", "wb");
    if (fp==NULL) {
        printf("Error: fopen() failed\n");
    }
}
```

Εγγραφή σε Δυναμικά αρχεία

```
        exit(EXIT_FAILURE);  
    }  
    for(i=0; str[i]!='\0'; i=i+1){  
        res=fwrite(&str[i], 1, 1, fp);  
        printf("RES= %d\n", res);  
    }  
    fclose(fp);  
    return 0;  
}
```

εγγράφει τα στοιχεία του `str` ανά χαρακτήρα στο αρχείο `test1.dat`.

- Η εντολή εγγραφής είναι καλύτερα να γράφεται ως εξής:
`fwrite(&str[i], sizeof(char), 1, fp);`
- Θα μπορούσαμε να εγγράψουμε την μεταβλητή και με τον παρακάτω τρόπο

Εγγραφή σε Δυναμικά αρχεία

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

int main() {
    char str[] = "programmatismos II";
    int res;
    FILE *fp;
    fp = fopen("test2.dat", "wb");
    if (fp == NULL) {
        printf("Error: fopen() failed\n");
        exit(EXIT_FAILURE);
    }
    res = fwrite(str, sizeof(str), 1, fp);
    fclose(fp);
    printf("RES= %d\n", res);
    return 0;
}
```

²προσοχή στην μεταβλητή αν είναι απλή ή πίνακας (δείκτης)

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Ανάγνωση σε Δυαδικά αρχεία

- Η συνάρτηση `fread()` διαβάζει δεδομένα από ένα δυαδικό αρχείο
- Το πρωτότυπο της συνάρτησης είναι

```
int fread(void *buf, size_t size, size_t count, FILE  
          *fp);
```

και η σύνταξη της εντολής είναι η εξής:

```
|| fread(arr, sizeof(int), 50, fp);
```

ή

```
|| fread(&a, sizeof(int), 1, fp);
```

η οποία δέχεται σαν ορίσματα

- ▶ την διεύθυνση μνήμης της μεταβλητής στην οποία θα καταχωρηθούν τα δεδομένα από το αρχείο,

Ανάγνωση σε Δυναμικά αρχεία

- ▶ το μέγεθος σε bytes για κάθε στοιχείου που θα διαβαστεί,
- ▶ τον αριθμό στοιχείων, το καθένα με μέγεθος bytes και
- ▶ το δείκτη αρχείου που έχει συσχετιστεί με ένα ανοικτό αρχείο.

Όταν η ανάγνωση είναι επιτυχής τότε επιστρέφει το πλήθος των αναγνώσεων που έχουν γίνει.

- Για παράδειγμα το παρακάτω πρόγραμμα

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

int main() {
    char str[18];
    int i, res;
    FILE *fp;
    fp = fopen("test1.dat", "rb");
    if (fp==NULL) {
        printf("Error: fopen() failed\n");
    }
}
```

Ανάγνωση σε Δυναμικά αρχεία

```
        exit(EXIT_FAILURE);
    }
    for(i=0; i<18; i=i+1){
        res=fread(&str[i], sizeof(char), 1, fp);
        printf("RES= %d\n", res);
    }
    fclose(fp);
    printf("%s\n", str);
    return 0;
}
```

διαβάζει ανά χαρακτήρα από το αρχείο `test1.dat` και τα καταχωρεί στο `str`.

- θα εμφανιστεί στην κονσόλα

Ανάγνωση σε Δυναμικά αρχεία

```
RES= 1  
.  
.  
.  
RES= 1  
programmatismos II
```

- Θα μπορούσαμε να διαβάσουμε από το αρχείο και με τον παρακάτω τρόπο

```
#include <stdio.h>  
#include <stdlib.h>  
  
int main() {  
    char str[18];  
    int res;  
    FILE *fp;  
    fp = fopen("test2.dat", "rb");
```

Ανάγνωση σε Δυναμικά αρχεία

```
if (fp==NULL) {  
    printf("Error: fopen() failed\n");  
    exit(EXIT_FAILURE);  
}  
res=fread(str, sizeof(str), 1, fp);  
fclose(fp);  
printf("RES= %d\n", res);  
printf("%s\n", str);  
return 0;  
}
```

διαβάζει ένα αλφαριθμητικό από το αρχείο `test2.dat` και το καταχωρεί στο `str`.

- θα εμφανιστεί στην κονσόλα

```
RES= 1  
programmatismos II
```

Δυαδικά αρχεία

Διαφορές μεταξύ δυαδικών αρχείων και αρχείων κειμένου.

- Για παράδειγμα το παρακάτω πρόγραμμα

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

int main() {
    int i;
    FILE *fpt, *fpb;
    fpt = fopen("diff.txt", "w");
    fpb = fopen("diff.dat", "wb");
    if ((fpt==NULL) || (fpb==NULL)) {
        printf("Error: fopen() failed\n");
        exit(EXIT_FAILURE);
    }
    for(i=0; i<1000; i=i+1) {
        fprintf(fpt, "%d\n", i);
```

Δυαδικά αρχεία

```
        fwrite(&i, sizeof(int), 1, fpb);  
    }  
    fclose(fpt);  
    fclose(fpb);  
    return 0;  
}
```

- το αρχείο `diff.txt` έχει μέγεθος 4890 bytes ενώ το αρχείο `diff.dat` έχει μέγεθος 4000 bytes.
- Ενώ στο παρακάτω πρόγραμμα στην ανάγνωση των αρχείων

Δυναμικά αρχεία

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

int main() {
    int i, arrT[1000], arrB[1000];
    FILE *fpt, *fpb;
    fpt = fopen("diff.txt", "rw");
    fpb = fopen("diff.dat", "rb");
    if ((fpt==NULL) || (fpb==NULL)) {
        printf("Error: fopen() failed\n");
        exit(EXIT_FAILURE);
    }
    for(i=0; i<1000; i=i+1) {
        fscanf(fpt, "%d", &arrT[i]);
    }
    fread(arrB, sizeof(int), 1000, fpb);
    fclose(fpt);
}
```

Δυναμικά αρχεία

```
    fclose(fpb);  
    printf("%d\n", arrT[500]);  
    printf("%d\n", arrB[500]);  
    return 0;  
}
```

- καλύτερη διαχείριση δεδομένων

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Αρχεία (Files)
- 3 Κοινές Λειτουργίες αρχείων
 - Άνοιγμα επεξεργασίας αρχείου
 - Κλείσιμο επεξεργασίας αρχείου
- 4 Εγγραφή και Ανάγνωση σε Αρχεία κειμένου
 - Εγγραφή σε Αρχεία κειμένου
 - Ανάγνωση σε Αρχεία κειμένου
- 5 Προσπέλαση Αρχείων
- 6 Εγγραφή και Ανάγνωση σε Δυναμικά αρχεία
 - Εγγραφή σε Δυναμικά αρχεία
 - Ανάγνωση σε Δυναμικά αρχεία
- 7 Παραδείγματα

Παράδειγμα 1: Αρχεία κειμένου

Να δημιουργηθεί πρόγραμμα το οποίο θα:

- θα εγγράφει τα ονόματα και τους βαθμούς φοιτητών σε ένα αρχείο κειμένου (σε στήλες)
- και έπειτα θα διαβάζει τους βαθμούς από το αρχείο και θα υπολογίζει και θα εμφανίζει το μέσο όρο βαθμών.

Παράδειγμα 2: Αρχεία κειμένου

Να δημιουργηθεί πρόγραμμα το οποίο θα:

- θα διαβάζει τα ονόματα (ανά γραμμή) από ένα αρχείο κειμένου
- θα εγγράφει στο ίδιο αρχείο (από την αρχή) τα ονόματα τα οποία ξεκινούν από το γράμμα "a"

Παράδειγμα 3: Δυαδικά αρχεία

Να δημιουργηθεί πρόγραμμα το οποίο από ένα δυαδικό αρχείο το οποίο περιέχει 1000 ακέραιους αριθμούς θα:

- διαβάζει και καταχωρεί σε αντίστοιχους πίνακες τους αριθμούς από τον 1 μέχρι τον 20 και από τον 951 μέχρι τον 970
- να υπολογίζει τον μέσο όρο κάθε πίνακα και να τους εμφανίζει

Παράδειγμα 3: Δυναμικά αρχεία - Λύση

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    FILE *pF;
    int arr1[20], arr2[20], i, sum1=0, sum2=0;
    pF=fopen("int1000.dat", "rb");
    fread(arr1, sizeof(int), 20, pF);
    fseek(pF, -50*sizeof(int), SEEK_END);
    fread(arr2, sizeof(int), 20, pF);
    fclose(pF);
    for(i=0; i<20; i=i+1) {
        sum1=sum1+arr1[i];
        sum2=sum2+arr2[i];
    }
    printf("MO1= %f\n", (float)sum1/20);
    printf("MO2= %f\n", (float)sum2/20);
    return 0;
}
```

Παράδειγμα 3: Δυαδικά αρχεία - Λύση

```
|| }
```

θα εμφανιστεί στην κονσόλα

```
|| MO1= 10.500000  
|| MO2= 960.500000
```

Παράδειγμα 4: Δυαδικά αρχεία

Να δημιουργηθεί πρόγραμμα το οποίο θα:

- δημιουργεί μια δομή (**struct**), **data** με δυο μέλη **code** και **description** τα οποία θα είναι τύπου ακέραιου και αλφαριθμητικού με 100 χαρακτήρες.
- ανοίγει ένα δυαδικό αρχείο με σκοπό την εγγραφή δεδομένων
- βάζει τιμές στα μέλη της δομής **d** και θα τα εγγράφει στο δυαδικό αρχείο
- ο βρόγχος επανάληψης θα τερματιστεί μετά τις πέντε (5) επαναλήψεις

Παράδειγμα 5: Δυναμικά αρχεία

Στο δυναμικό αρχείο `myfile.ddd` έχουμε αποθηκεύσει τους ακέραιους αριθμούς 200, 201, ..., 300. Δίνεται το παρακάτω πρόγραμμα

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    FILE *pF;
    int a,b,c;
    pF=fopen("myfile.ddd","rb");
    fseek(pF,0*sizeof(int),SEEK_CUR);
    fread(&a,sizeof(int),1,pF);
    fseek(pF,-1*sizeof(int),SEEK_END);
    fread(&b,sizeof(int),1,pF);
    fseek(pF,10*sizeof(int),SEEK_SET);
    fread(&c,sizeof(int),1,pF);
    fclose(pF);
    printf("%d %d %d\n",a,b,c);
}
```


Παράδειγμα 5: Δυναδικά αρχεία

```
|| return 0;
```

```
|| }
```

- Να γράψετε τις τιμές των μεταβλητών όπως εμφανίζονται στο παραπάνω πρόγραμμα.

Παράδειγμα 5: Δυαδικά αρχεία - Λύση

Θα εμφανιστεί στην κονσόλα

```
|| 200 300 210
```