



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

Εισαγωγή στον δομημένο προγραμματισμό

Ενότητα 11^η: Αρχεία

Αν. καθηγητής Στεργίου Κώστας
e-mail: kstergiou@uowm.gr

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών



Πανεπιστήμιο Δυτικής Μακεδονίας



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Στόχοι της διάλεξης

- Περιγραφή του τρόπου χρήσης των αρχείων από τη C για την αποθήκευση δεδομένων στη δευτερεύουσα μνήμη.
- Εξοικείωση με τις βασικές τεχνικές χρήσης αρχείων.
- Ανάλυση των διαφορών μεταξύ αρχείων κειμένου και δυαδικών αρχείων.



Περιεχόμενα

- Αρχεία.
 - Δήλωση, άνοιγμα, κλείσιμο αρχείου.
- Αρχεία κειμένου.
 - Εγγραφή σε αρχεία κειμένου.
 - Ανάγνωση από αρχεία κειμένου.
- Δυαδικά αρχεία.
 - Εγγραφή σε δυαδικά αρχεία.
 - Ανάγνωση από δυαδικά αρχεία.
 - Τυχαία προσπέλαση.



Αρχεία (1/2)

- Συλλογές δεδομένων.
- Αποθηκεύονται στην περιφερειακή μνήμη.
 - π.χ. σκληρός δίσκος.
- Από τη στιγμή δημιουργίας τους τα δεδομένα. που περιέχουν διατηρούνται όσο είναι επιθυμητό.
 - Αντίθετα, ένας πίνακας διατηρείται όσο εκτελείται ένα πρόγραμμα.



Αρχεία (2/2)

- Ανάλογα με τον τρόπο αποθήκευσης διακρίνονται σε:
 - Αρχεία κειμένου: δεδομένα ως αναγνώσιμοι και εκτυπώσιμοι χαρακτήρες.
 - Δυαδικά αρχεία: δεδομένα με δυαδική αναπαράσταση.
- Τα ίδια δεδομένα μπορούν να αποθηκευτούν και με τις δύο μορφές.



Αρχεία και Ρεύματα

- Για την χρήση των αρχείων είναι απαραίτητη η δημιουργία ενός ρεύματος (stream).
- Το ρεύμα είναι η σύνδεση ενός προγράμματος με ένα αρχείο.
- Τα κύρια ρεύματα είναι:
 - Το βασικό ρεύμα εισόδου (stdin).
 - Το βασικό ρεύμα εξόδου (stdout).



Διαδικασία Δημιουργίας Αρχείου

- Για τη δημιουργία ενός αρχείου απαιτείται η ακόλουθη διαδικασία:
 1. Δήλωση μιας μεταβλητής δείκτη FILE (δημιουργία ρεύματος) για τη διαχείριση του αρχείου.
 2. Άνοιγμα του αρχείου με χρήση της συνάρτησης fopen.
 3. Επεξεργασία των δεδομένων του αρχείου.
 4. Κλείσιμο του αρχείου με χρήση της συνάρτησης fclose..



Δήλωση Αρχείων

- Δημιουργία ενός ρεύματος που συνδέει ένα πρόγραμμα με ένα αρχείο.
 - FILE *<όνομα1>,<όνομα2>,...,*<όνομαN>;
 - FILE είναι μια δομή αρχείου ορισμένη στο stdio.h.
 - <όνομαX> είναι ένας δείκτης και αντιστοιχεί στο εσωτερικό όνομα αρχείου.
- Το εσωτερικό όνομα ενός αρχείου χρησιμοποιείται για τη διαχείριση του αρχείου εντός του προγράμματος.
- Το εξωτερικό όνομα ενός αρχείου χρησιμοποιείται από το λειτουργικό σύστημα.
 - π.χ. C:\MYFILES\FILE1.TXT



Άνοιγμα Αρχείων

- Πριν γίνει οποιαδήποτε επεξεργασία σε ένα αρχείο πρέπει πρώτα να ανοιχθεί.
- Στο άνοιγμα συνδέεται το εσωτερικό όνομα του αρχείου με το εξωτερικό.
- <εσωτερικό όνομα> = fopen(<εξωτερικό όνομα>, <τρόπος>);
- Ο <τρόπος> δηλώνει τον τύπο της επεξεργασίας που θα υποστεί το αρχείο.



Τρόποι Ανοίγματος

- "r" :
διάβασμα δεδομένων από το αρχείο.
- "w" :
γράψιμο δεδομένων στο αρχείο (καταστρέφει τα δεδομένα αν το αρχείο ήδη υπάρχει).
- "a" :
προσθήκη δεδομένων στο τέλος του ήδη υπάρχοντος αρχείου.
- "r+" :
ενημέρωση των δεδομένων του αρχείου (διάβασμα και γράψιμο δεδομένων).



Παράδειγμα 1^ο

```
#include <stdio.h>
#define arxeio1 "c:\\arxeio1.txt"
    /* ΠΡΟΣΟΧΗ στο διπλό '\\' */

void main ()
{
    FILE *myFile1, *myFile2, *myFile3;
    char arxeio2[16] = "c:\\arxeio2.txt";

    myFile1 = fopen(arxeio1, "r");
    myFile2 = fopen(arxeio2, "w");
    myFile3 = fopen("c:\\arxeio3.txt", "a");
}
```



Αποτυχία fopen

- Αν ένα όνομα αρχείου δεν υπάρχει, η fopen θα αποτύχει και επιστρέφει τιμή NULL.

```
myFile3 = fopen("c:\\arxeio3.txt", "w");  
if (!myFile3)  
    printf("Error opening \"%myFile3\"\n");  
else  
    printf("\"myFile3\" opened successfully\n");
```



Κλείσιμο Αρχείων

- Μετά την επεξεργασία των δεδομένων από το αρχείο συνιστάται να το κλείνουμε.
- Το κλείσιμο του αρχείου δεν είναι απόλυτα αναγκαίο.
- Το λειτουργικό σύστημα κλείνει τα ανοικτά αρχεία μετά τον τερματισμό του προγράμματος.
- `fclose (<εσωτερικό όνομα>);`
 - Αποσυνδέεται το αρχείο <εσωτερικό όνομα> από το πρόγραμμα.
- Μετά το κλείσιμο ενός αρχείου, μπορούμε να το ξαναανοίξουμε με άλλο τρόπο ανοίγματος.



Παράδειγμα 2^ο

```
#define arxeio1 "c:\\arxeio1.txt"
void main ()
{
    FILE *myFile1, *myFile2, *myFile3;
    char arxeio2[16] = "c:\\arxeio2.txt";

    myFile1 = fopen(arxeio1, "r");
    myFile2 = fopen(arxeio2, "w");
    myFile3 = fopen("c:\\arxeio3.txt", "a");
    /* ... εντολές επεξεργασίας δεδομένων */
    fclose(myFile1);
    fclose(myFile2);
    fclose(myFile3);
}
```



Η Συνάρτηση feof

- Η συνάρτηση feof χρησιμοποιείται για τον εντοπισμό του τέλους ενός αρχείου (**end of file**).
- `feof (<εσωτερικό όνομα>);`
- Είναι αληθής (επιστρέφει τιμή διάφορη του μηδενός) όταν επιχειρείται να γίνει ανάγνωση του αρχείου μετά το σημάδι τέλους του αρχείου.
- Είναι ψευδής (επιστρέφει μηδέν) όταν η τρέχουσα θέση ανάγνωσης του αρχείου είναι πριν το σημάδι τέλους του αρχείου .
- Περιλαμβάνεται στη βιβλιοθήκη `<stdio.h>`



Παράδειγμα: feof

```
#include <stdio.h>
void main()
{
    FILE *myStream;
    if (!(myStream = fopen( "c:\\test.txt", "r")))
        printf("Error opening the file");
    /* Epanalipsi mexri to telos tou arxeiou: */
    while (!feof(myStream))
    {
        feof(myStream) == NULL

        /* diabase 10 bytes tou arxeiou test.txt */
        /* Athrise poses dekades diabastikan */
    }
    /* ektypose poses dekades bytes diabastikan */
    fclose(myStream);
}
```



Αρχεία Κειμένου

- Τα περιεχόμενα των αρχείων κειμένου είναι αναγνώσιμοι και εκτυπώσιμοι χαρακτήρες.
- Κάθε εγγραφή διαχωρίζεται από την επόμενη με τον χαρακτήρα νέας γραμμής "`\n`".
- Για το διάβασμα και το γράψιμο δεδομένων χρησιμοποιούνται οι ίδιες συναρτήσεις με τη διαλογική επεξεργασία.
 - Αντί των βασικών ρευμάτων `stdin` και `stdout` χρησιμοποιούνται τα αρχεία που δίνονται ως ορίσματα για την είσοδο και την έξοδο δεδομένων.



Εγγραφή σε Αρχείο Κειμένου

- Η εγγραφή οποιουδήποτε βασικού τύπου δεδομένων (char, float, int, double) επιτυγχάνεται με τη χρήση της fprintf (παραλλαγή της printf).
- printf(“συμβολοσειρά”, [ορίσματα]);
- fprintf(<εσ.όνομα>, “συμβολοσειρά”, [ορίσματα]);
- Άλλες συναρτήσεις:
 - fputc(<χαρακτήρας>, <εσ.όνομα>);
 - fputs(<αλφαριθμητικό>, <εσ.όνομα>);



Παράδειγμα 3^ο

```
/* Εκτύπωση σε αρχείο των αριθμών και των
   αντίστοιχων χαρακτήρων ASCII από 0 έως 99 */

#include <stdio.h>
void main()
{
    FILE *myStream;
    if (!(myStream = fopen( "c:\\test.txt", "w")))
        printf("Error opening the file \"test.txt\"");
    else
    {
        for (int j=0; j < 100; j++)
            fprintf(myStream, "%d:%c\n", j, j);
        fclose(myStream);
    }
}
```



Ανάγνωση Αρχείων Κειμένου

- Αφού ανοιχθεί ένα αρχείο (για ανάγνωση ή ανάγνωση/εγγραφή) χρησιμοποιούνται οι εντολές διαβάσματος αρχείου:
- `fscanf(<εσ.όνομα>, “συμβολοσειρά”, [ορίσματα]);`
- `fgetc(<εσ.όνομα>);`
- `fgets(<αλφαριθμητικό>, <μέγεθος>, <εσ.όνομα>);`
 - Προσοχή η `fgets` διαβάζει αλφαριθμητικά με πλήθος χαρακτήρων το πολύ `<μέγεθος>-1`.



Παράδειγμα 4^ο

```
/* Διάβασμα εγγραφών ακεραίων, εκτύπωση αθροίσματός τους και πλήθους εγγραφών */
#include <stdio.h>
#define arxeio1 "c:\\test.txt"
void main ()
{
    FILE *arxeio;
    int arithmos,athroisma,plithos;
    athroisma=0;
    plithos=0;
    if (!(arxeio=fopen(arxeio1,"r"))) {
        printf("Error opening file: %s",arxeio1);
        return; }
    while (!feof(arxeio)) {
        fscanf(arxeio,"%d\n",&arithmos);
        athroisma+=arithmos;
        plithos++; }
    fclose(arxeio);
    printf("Plithos:%d\n", plithos) ;
    printf("Athroisma:%d\n", athroisma) ;
}
```



Ανάγνωση Μορφοποιημένων Αρχείων (1/4)

- Πιθανώς μια εγγραφή δεν είναι επιθυμητό να διαβαστεί αυτούσια αλλά ένα συγκεκριμένο μέρος της.
- Είναι δυνατό να χρησιμοποιήσουμε προσδιοριστές μορφής με παράγοντες πλάτους.
- π.χ., για την εγγραφή 14.3515.6517.85.



Ανάγνωση Μορφοποιημένων Αρχείων (2/4)

- Επιθυμούμε να καταχωρηθούν οι τιμές 14.351, 5.65 και 17.85 στις μεταβλητές x , y , z .
- Χρησιμοποιούμε τον προσδιοριστή μορφής `%wf` όπου w είναι το πλάτος του πεδίου που καταλαμβάνει η επιθυμητή τιμή.
- `fscanf (eisodos, "%6f%4f%5f", &x, &y, &z) ;`



Ανάγνωση Μορφοποιημένων Αρχείων (3/4)

- Οι τιμές μιας εγγραφής μπορεί να χωρίζονται από έναν ή περισσότερους αδιάφορους σε εμάς χαρακτήρες. Τότε μπορούμε να κάνουμε τα εξής:
 1. Να μεταφέρουμε τους χαρακτήρες αυτούς σε μεταβλητές τις οποίες στη συνέχεια δε θα χρησιμοποιήσουμε.
 2. Να εισάγουμε τους αδιάφορους χαρακτήρες στο προσδιοριστικό μορφής.
 3. Να χρησιμοποιήσουμε το χαρακτήρα καταπίεσης καταχώρησης "*" .



Ανάγνωση Μορφοποιημένων Αρχείων (4/4)

Παράδειγμα εισόδου: `abcd, 50, 60, 70.`

```
1. fscanf(eisodos, "%4s%c%d%c%d%c%d", onoma,  
    &comma, &int1, &comma, &int2, &comma,  
    &int3);
```

```
2. fscanf(eisodos, "%4s,%d,%d,%d", onoma,  
    &int1, &int2, &int3);
```

```
3. fscanf(eisodos, "%s%*c%d%*c%d%*c%d",  
    onoma, &int1, &int2, &int3);
```



Παράδειγμα: Αντιγραφή Αρχείων (1/2)

```
#include <stdio.h>

void antigrafi(char source[], char destination[]);

void main()
{
    char source[100];
    char destination[100];
    printf("Enter source file name\n");
    gets(source);
    printf("Enter destination file name\n");
    gets(destination);
    antigrafi(source, destination);
}
```



Παράδειγμα: Αντιγραφή Αρχείων (2/2)

```
void antigrafi(char source[], char destination[])
{
    char ch;
    int counter = 0;
    FILE *sour, *dest;
    if (!(sour = fopen(source, "r")))
        printf("Error opening the file %s", source);
    if (!(dest = fopen(destination, "w")))
        printf("Error opening the file %s", destination);
    while (!feof(sour))
    {
        ch = fgetc(sour);
        fputc(ch, dest);
        counter++;
    }
    printf("%d characters copied\n", counter);
    fclose(sour);
    fclose(dest);
}
```



Δυαδικά Αρχεία (1/2)

- Σε αντίθεση με τα αρχεία κειμένου, τα περιεχόμενα των δυαδικών αρχείων είναι αναγνώσιμα μόνο από τη μηχανή.
- Τα δυαδικά αρχεία απαιτούν λιγότερη μνήμη από τα αρχεία κειμένου.
- Τα δυαδικά αρχεία είναι δυνατό να αποθηκεύσουν δομημένες εγγραφές.
- Η C υποστηρίζει δυαδικά αρχεία μέσω των συναρτήσεων `fwrite` και `fread`.



Δυαδικά Αρχεία (2/2)

- Οι υπόλοιπες εντολές και συναρτήσεις είναι οι ίδιες με τα αρχεία κειμένου.
- Επεξεργασία ενός δυαδικού αρχείου:
 1. Δήλωση εσωτερικού ονόματος του δυαδικού αρχείου.
 2. Άνοιγμα του αρχείου με τη συνάρτηση `fopen`.
 3. Επεξεργασία του αρχείου.
 4. Κλείσιμο του αρχείου με κλήση της συνάρτησης `fclose`.



Δημιουργία Δυαδικών Αρχείων

- Για το γράψιμο μιας εγγραφής σε ένα δυαδικό αρχείο χρησιμοποιείται η συνάρτηση `fwrite`.
- `fwrite(<διεύθυνση>, <μέγεθος>, <πλήθος>, <εσωτερικό όνομα>);`
 - <διεύθυνση> είναι η διεύθυνση μιας ενδιάμεσης προσωρινής περιοχής αποθήκευσης (buffer) στην κύρια μνήμη, η οποία θεωρείται ως πίνακας.
 - <μέγεθος> είναι ο ακέραιος ο οποίος παριστάνει τ.ο μήκος των bytes της ενδιάμεσης προσωρινής περιοχής αποθήκευσης.
 - <πλήθος> είναι το πλήθος των στοιχείων του πίνακα της ενδιάμεσης προσωρινής περιοχής αποθήκευσης (εάν η ενδιάμεση προσωρινή περιοχή αποθήκευσης δεν είναι πίνακας τότε το πλήθος είναι ίσο με τη μονάδα).
 - <εσωτερικό όνομα> είναι το εσωτερικό όνομα του αρχείου που θα εγγραφούν τα δεδομένα.
- Η συνάρτηση `fwrite` επιστρέφει το πλήθος των στοιχείων του πίνακα που γράφτηκαν επιτυχώς.



Παράδειγμα 5^ο

```
FILE *exodos;
```

```
int num;
```

```
exodos = fopen("C:\\TEST.TXT", "w");
```

```
fwrite(&num, sizeof(num), 1, exodos);
```

```
/* Αποθηκεύονται 2 ή 4 bytes, ανάλογα με το  
μηχάνημα, που διαβάζονται από τη διεύθυνση  
μνήμης num */
```



Παράδειγμα 6^ο

```
#include <stdio.h>

void main()
{
    FILE *arxeioA;
    FILE *arxeioB;
    FILE *arxeioC;
    char seira1[20];
    int seira2[5];
    int akeraios = 33;
    arxeioA = fopen("arxeio1.txt", "w");
    arxeioB = fopen("arxeio2.txt", "w");
    arxeioC = fopen("arxeio3.txt", "w");

    fwrite(seira1, (sizeof(char)*20), 1, arxeioA);
    fwrite(seira2, (sizeof(int)*5), 1, arxeioB);
    fwrite(&akeraios, sizeof(int), 1, arxeioC);
}
```



Ανάγνωση Δυαδικών Αρχείων

- Την ανάγνωση δεδομένων από ένα ήδη υπάρχον δυαδικό αρχείο αναλαμβάνει η συνάρτηση `fread`.
- `fread(<διεύθυνση>, <μέγεθος>, <πλήθος>, <εσωτερικό όνομα>);`
- Η ομοιότητα της `fread` με την `fwrite` είναι πλήρης εκτός από το ότι στην `fread` μεταφέρονται bytes από το αρχείο προς τη διεύθυνση μιας μεταβλητής.



Παράδειγμα 7^ο

```
FILE *eisodos;
```

```
int num;
```

```
eisodos = fopen("C:\\TEST.TXT", "r");
```

```
fread(&num, sizeof(num), 1, eisodos);
```

```
/* Διαβάζονται 2 ή 4 bytes, ανάλογα με το μηχάνημα  
και μεταφέρονται στη διεύθυνση μνήμης num */
```



Παράδειγμα 8^ο

```
/* Διάβασμα ακεραίου από δυαδικό αρχείο */  
  
#include <stdio.h>  
void main()  
{  
    FILE *arxeioA;  
    int temp;  
    arxeioA = fopen("arxeio1.txt", "r");  
    while (!feof(arxeioA)) {  
        fread(&temp, sizeof(temp), 1, arxeioA);  
        printf("Periexomeno arxeiou:%d\n", temp);  
    }  
    fclose(arxeioA);  
}
```



Διαχωρισμός/Συγχώνευση Αρχείων

- Ο διαχωρισμός ενός δυαδικού αρχείου σε δύο ή περισσότερα αρχεία γίνεται με βάση ένα μέλος-κλειδί των εγγραφών του.
 - Επεξεργάζονται όλες οι εγγραφές του αρχείου ξεκινώντας από την πρώτη.
 - Με κριτήριο την τιμή ενός μέλους-κλειδί κάθε εγγραφή τοποθετείται στο αντίστοιχο νέο αρχείο.
- Η συγχώνευση δύο ή περισσότερων αρχείων είναι η αντίστροφη διαδικασία.



Παράδειγμα: Διαχωρισμός (1/2)

- Δίνεται ένα δυαδικό αρχείο που περιέχει πίνακα ακεραίων με 20 στοιχεία με πεδίο τιμών $[0,40]$.
- Τα στοιχεία του πίνακα θα τα μοιράσουμε σε 3 αρχεία τα περιεχόμενα των οποίων θα εκτυπωθούν στην οθόνη.
 - Το πρώτο αρχείο θα περιέχει τιμές $[0,9]$.
 - Το δεύτερο θα περιέχει τιμές $[10,19]$.
 - Το τρίτο θα περιέχει τιμές $[20, 40]$.



Παράδειγμα: Διαχωρισμός (2/2)

```
arxeio1 = fopen("arxeio1.txt", "r");
arxeio2 = fopen("arxeio_proor1.txt", "w+");
arxeio3 = fopen("arxeio_proor2.txt", "w+");
arxeio4 = fopen("arxeio_proor3.txt", "w+");

printf("Το περιεχόμενο του αρχείου είναι:");
int temp;
while (!feof(arxeio1))
{
    if (fread(&temp, sizeof(temp), 1, arxeio1))
        printf("%d ", temp);
    if (temp <= 9)
        fwrite(&temp, sizeof(temp), 1, arxeio2);
    else if (temp <= 19)
        fwrite(&temp, sizeof(temp), 1, arxeio3);
    else
        fwrite(&temp, sizeof(temp), 1, arxeio4);
}
printf("\n");
fclose(arxeio1); fclose(arxeio2);
fclose(arxeio3); fclose(arxeio4);
```



Τυχαία Προσπέλαση (1/2)

- Για κάθε αρχείο που ανοίγει η C δημιουργείται ένας δείκτης (γραφής ή ανάγνωσης) ώστε να είναι δυνατή η ακολουθιακή προσπέλαση του αρχείου.
 - Η προσπέλαση του N -στού στοιχείου βασίζεται στα προηγούμενα $N-1$ στοιχεία που προσπελάσθηκαν.
- Σε αντίθεση με την ακολουθιακή προσπέλαση, η τυχαία επιτρέπει την προσπέλαση του N -στού στοιχείου χωρίς να είναι απαραίτητη η προσπέλαση των υπολοίπων $N-1$ στοιχείων.
- Στη C, η τυχαία προσπέλαση υποστηρίζεται μέσω των συναρτήσεων `ftell` και `fseek` (`stdio.h`).



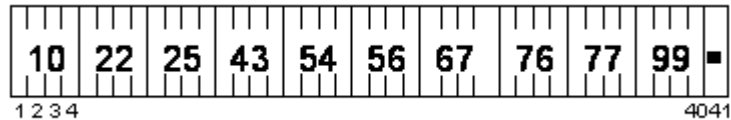
Τυχαία Προσπέλαση (2/2)

- `ftell(<εσωτερικό όνομα>);`
 - Εντοπίζει την τρέχουσα τιμή του δείκτη του αρχείου και επιστρέφει τον αριθμό των bytes από την αρχή.
- `fseek(<εσωτερικό όνομα>, <απόσταση>, <θέση>);`
 - Μετακινεί τον δείκτη κατά ένα αριθμό bytes.
 - <απόσταση> η απόσταση που θα μετακινηθεί ο δείκτης σε bytes.
 - <θέση> είναι μια από τις ακόλουθες τρεις σταθερές:
 - `SEEK_SET`: μετακινεί το δείκτη <απόσταση> bytes από την αρχή του ρεύματος του αρχείου.
 - `SEEK_END`: μετακινεί το δείκτη <απόσταση> bytes από το τέλος του ρεύματος του αρχείου.
 - `SEEK_CUR`: μετακινεί το δείκτη <απόσταση> bytes από την τρέχουσα θέση του .



Παράδειγμα 9^ο (1/2)

- θεωρούμε ότι ένας ακέραιος απαιτεί χώρο 4 bytes (γεγονός που εξαρτάται από το λειτουργικό σύστημα).
- Έχουμε ένα δυαδικό αρχείο με εγγραφές ακεραίων όπως στο ακόλουθο σχήμα.



- Για κάποιο λόγο ο χρήστης επιθυμεί ΜΟΝΟ τις ζυγές εγγραφές διαβασμένες από το τέλος.
- Με χρήση των `ftell` και `fseek` είναι πολύ εύκολο να προσπελαστούν οι εγγραφές αυτές χωρίς ακολουθιακή προσπάθεια του αρχείου.



Παράδειγμα 9^ο (2/2)

```
#include <stdio.h>
void main() {
int int_size = sizeof(int);
FILE *arxeio;
    int seira_akeraiou[10];
int temp_akeraios, j;
    arxeio = fopen("arxeio.txt", "w+");
    seira_akeraiou = {10,22,25,43,54,56,67,76,77,99};
    fwrite(seira_akeraiou, (int_size*10), 1, arxeio);
    fclose(arxeio);
    arxeio = fopen("arxeio.txt", "r");
    fseek(arxeio, -int_size, SEEK_END);
    printf("Οι επιθυμητές εγγραφές του \"arxeio.txt\" είναι:");
for (j=0; j<5; j++)
{
    if (fread(&temp_akeraios, sizeof(temp_akeraios), 1, arxeio))
        printf("%d ", temp_akeraios);
        fseek(arxeio, -3*int_size, SEEK_CUR);
}
printf("\n");
fclose(arxeio);
}
```



Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Σημείωμα Αναφοράς

- Copyright Πανεπιστήμιο Δυτικής Μακεδονίας, Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών, Στεργίου Κωνσταντίνος. «Εισαγωγή στον Δομημένο Προγραμματισμό». Έκδοση: 1.0. Κοζάνη 2015. Διαθέσιμο από τη δικτυακή διεύθυνση: <https://eclass.uowm.gr/courses/ICTE258/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Όχι Παράγωγα Έργα Μη Εμπορική Χρήση 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Ως Μη Εμπορική ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό



Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

