



Προγραμματισμός Υπολογιστών & Υπολογιστική Φυσική

Ενότητα 8: Αρχεία και Δομές

Νικόλαος Στεργιούλας
Τμήμα Φυσικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

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 Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την 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.



ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ & ΥΠΟΛΟΓΙΣΤΙΚΗ ΦΥΣΙΚΗ

Μέρος 8ο

ΝΙΚΟΛΑΟΣ ΣΤΕΡΓΙΟΥΛΑΣ



ΤΜΗΜΑ ΦΥΣΙΚΗΣ

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΡΧΕΙΑ ΚΕΙΜΕΝΟΥ (ASCII TEXT FILES)

Τα αρχεία κειμένου περιέχουν **χαρακτήρες** σύμφωνα με την κωδικοποίηση **ASCII**.

Κάθε γραμμή τελειώνει με έναν ειδικό χαρακτήρα (**τέλος γραμμής**) ο οποίος είναι δεν είναι ορατός στους συνηθισμένους επεξεργαστές κειμένου.

Στο λειτουργικό σύστημα **Windows** το τέλος γραμμής εκφράζεται με το συνδυασμό **\r\n** (return και line feed), ενώ στο **Unix** μόνο με **\n**.

ΔΥΑΔΙΚΑ ΑΡΧΕΙΑ (BINARY FILES)

Τα **δυναδικά αρχεία** δεν περιέχουν αναγνώσιμους χαρακτήρες, αλλά η πληροφορία αποθηκεύεται σε **ειδική ψηφιακή (δυναδική) μορφή**.

Για παράδειγμα, το εκτελέσιμο αρχείο ενός προγράμματος είναι σε δυναδική μορφή και δεν μπορούμε να δούμε τα περιεχόμενά του με έναν επεξεργαστή κειμένου (υπάρχουν ειδικά προγράμματα για την αποκωδικοποίηση και ανάγνωση τέτοιων αρχείων).

Τα δυναδικά αρχεία έχουν **αρκετά μικρότερο μέγεθος** από ένα αντίστοιχο αρχείο κειμένου, διότι κάθε χαρακτήρας αποθηκεύεται με τον ελάχιστο αριθμό bytes που απαιτούνται.

ΠΑΡΑΔΕΙΓΜΑ ΜΕΓΕΘΟΥΣ ΑΡΧΕΙΩΝ

Το μέγεθος ενός αρχείου κειμένου που περιέχει τον αριθμό **47654** είναι **5 bytes** (ένα byte για κάθε ψηφίο).

Ο αριθμός αυτός στο δυαδικό σύστημα είναι ο **1011101000100110** και απαιτεί 16 bit = **2 bytes** για να αποθηκευτεί.

					Ο ASCII του κάθε ψηφίου
<u>Αρχείο κειμένου:</u>	00110100	00110111	00110110	00110101	00110100
	'4'	'7'	'6'	'5'	'4'
<u>Δυαδικό αρχείο:</u>	10111010	00100110			

ΑΝΟΙΓΜΑ ΑΡΧΕΙΩΝ

Για το άνοιγμα ενός αρχείου χρησιμοποιούμε την συνάρτηση `fopen` που είναι δηλωμένη στο `stdio.h`.

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

όπου

`filename` = το όνομα του αρχείου, π.χ. "mytext1.txt" (όταν βρίσκεται στον ίδιο φάκελο με το εκτελέσιμο αρχείο)

`mode` = χαρακτήρες που καθορίζουν τις ενέργειες που μπορούν να γίνουν στο αρχείο, π.χ. `r` (βλ. πίνακα στη συνέχεια)

ΕΝΕΡΓΕΙΕΣ ΣΕ ΑΡΧΕΙΑ

Οι πιθανές τιμές που μπορεί να πάρει η μεταβλητή **mode** και οι **ενέργειες** στις οποίες αντιστοιχούν αυτές φαίνονται στον παρακάτω πίνακα:

Επιλογή	Ενέργεια
r	Ανοίγει το αρχείο για ανάγνωση.
w	Ανοίγει το αρχείο για εγγραφή. Αν το αρχείο δεν υπάρχει, δημιουργείται. Αν υπάρχει, το περιεχόμενό του διαγράφεται.
a	Ανοίγει το αρχείο για προσάρτηση. Αν το αρχείο δεν υπάρχει, δημιουργείται. Αν υπάρχει, τα υπάρχοντα δεδομένα διατηρούνται και τα νέα δεδομένα προστίθενται στο τέλος.
r+	Ανοίγει το αρχείο για ανάγνωση και εγγραφή.
w+	Ανοίγει το αρχείο για ανάγνωση και εγγραφή. Αν το αρχείο δεν υπάρχει, δημιουργείται. Αν υπάρχει, το περιεχόμενό του διαγράφεται.
a+	Ανοίγει το αρχείο για ανάγνωση και προσάρτηση. Αν το αρχείο δεν υπάρχει, δημιουργείται. Αν υπάρχει, τα υπάρχοντα δεδομένα διατηρούνται και τα νέα δεδομένα προστίθενται στο τέλος.

Εάν πρόκειται για δυαδικό αρχείο, προστίθεται ο χαρακτήρας **b**, π.χ. αντί για **r** γράφουμε **rb**.

ΕΠΙΤΥΧΙΑ/ΑΠΟΤΥΧΙΑ ΑΝΟΙΓΜΑΤΟΣ ΑΡΧΕΙΩΝ

Εάν πράγματι ανοίξει το αρχείο με επιτυχία, τότε η `fopen` επιστρέφει έναν δείκτη σε μια δομή τύπου `FILE`.

Εάν αποτύχει το άνοιγμα του αρχείου (για παράδειγμα, εάν το αρχείο δεν υπάρχει), τότε επιστρέφεται η τιμή `NULL`.

Προς αποφυγή μη-επιτρεπτών ενεργειών όταν το αρχείο δεν υπάρχει, θα πρέπει να ελέγχουμε την ύπαρξή του ενσωματώνοντας την `fopen` μέσα σε έναν έλεγχο `if`, ως εξής:

```
if ( (fp = fopen("mytext.txt", "r")) == NULL)
```

ΚΛΕΙΣΙΜΟ ΑΡΧΕΙΟΥ

Όταν ολοκληρώσουμε την επεξεργασία ενός αρχείου, πρέπει να το κλείσουμε με την εντολή `fclose`, που ορίζεται στο `stdio.h` ως εξής:

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

όπου `fp` είναι ο δείκτης που αντιστοιχεί στο αρχείο που ανοίξαμε. Η `fclose` επιστρέφει `0` εάν το αρχείο έκλεισε επιτυχώς, αλλιώς επιστρέφει την ειδική τιμή `EOF` (που έχει τιμή `-1`).

Η `EOF` χρησιμεύει επίσης για να δηλώσει το τέλος ενός αρχείου (End Of File).

Η ΣΥΝΑΡΤΗΣΗ `fprintf`

Για να γράψουμε δεδομένα σε ένα αρχείο κειμένου χρησιμοποιούμε την συνάρτηση `fprintf` που είναι δηλωμένη ως εξής:

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

δηλαδή είναι παρόμοια με την `printf`, αλλά υπάρχει **ένα επιπλέον όρισμα στην αρχή**, που δείχνει στο αρχείο που ανοίγουμε.

Η `fprintf` επιστρέφει τον **αριθμό των χαρακτήρων που γράψαμε στο αρχείο**, αλλιώς μια αρνητική τιμή.

Η ΣΥΝΑΡΤΗΣΗ `fscanf`

Για να διαβάσουμε δεδομένα από ένα αρχείο κειμένου χρησιμοποιούμε την συνάρτηση `fscanf` που είναι δηλωμένη ως εξής:

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

δηλαδή είναι παρόμοια με την `scanf`, αλλά υπάρχει **ένα επιπλέον όρισμα στην αρχή**, που δείχνει στο αρχείο που ανοίγουμε.

Η `fscanf` επιστρέφει τον **αριθμό των στοιχείων που διαβάστηκαν** και εκχωρήθηκαν σε μεταβλητές του προγράμματος. Αν φτάσουμε στο τέλος του αρχείου χωρίς να διαβάσει κάποια στοιχεία ή αν συμβεί κάποιο σφάλμα, επιστρέφει EOF.

Η ΣΥΝΑΡΤΗΣΗ `fscanf`

Για να χρησιμοποιήσουμε την συνάρτηση `fscanf` πρέπει απαραίτητως να γνωρίζουμε επακριβώς τη μορφή των δεδομένων που βρίσκονται στο αρχείο που διαβάζουμε, ώστε να επιλέξουμε την κατάλληλη μορφοποίηση.

Οι τιμές μεταξύ των δεδομένων διαχωρίζονται με ένα κενό.

ΔΟΜΕΣ - ΟΡΙΣΜΟΣ ΠΡΟΤΥΠΟΥ

Μια **δομή** (**structure**) είναι η **ομαδοποίηση ενός αριθμού μεταβλητών οποιουδήποτε τύπου**.

Αρχικά, πρέπει να **οριστεί** το **πρότυπο** μιας δομής, το οποίο **καθορίζει το πλήθος και τον τύπο των μεταβλητών που αυτή θα περιέχει**. Η γενική μορφή του ορισμού του προτύπου μιας δομής είναι:

```
struct όνομα_προτύπου  
{  
    τύπος μεταβλητή_1;  
    τύπος μεταβλητή_2;  
    τύπος μεταβλητή_3;  
    ...  
    τύπος μεταβλητή_N;  
};
```

```
struct date
{
    int day;
    int month;
    int year;
};
```

```
struct person
{
    char name[50];
    int age;
    float height;
};
```


ΔΗΛΩΣΗ ΔΟΜΩΝ

Εφόσον έχουμε πρώτα ορίσει το **πρότυπο** μιας δομής (αυτό μπορεί να γίνει και πριν από την `main`, ώστε να έχει καθολική ισχύ για το πρόγραμμα) μπορούμε στη συνέχεια να **δηλώσουμε δομές** που ακολουθούν το συγκεκριμένο πρότυπο. Αυτό γίνεται με μια δήλωση της μορφής:

```
struct όνομα_προτύπου όνομα_δομής_1, όνομα_δομής_2, ...;
```

π.χ.

```
struct date date1, date2;
```

```
struct person person1, person2;
```

ΕΝΑΛΛΑΚΤΙΚΟΣ ΤΡΟΠΟΣ ΔΗΛΩΣΗΣ ΔΟΜΩΝ

Η δήλωση μιας δομής μπορεί να γίνει (εναλλακτικά) ταυτόχρονα με τον ορισμό του προτύπου.

π.χ.

```
struct person
{
    char name[50];
    int age;
    float height;
} person1, person2;
```

Μπορούμε να δώσουμε τιμές στα διάφορα πεδία της δομής ως εξής: π.χ. `person1.age = 23` .

ΠΑΡΑΔΕΙΓΜΑ

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

struct book
{
    char title[100];
    int year;
    float price;
};

int main(void)
{
    struct book book1;

    strcpy( book1.title, "Programming in C");
    book1.year = 2015;
    book1.price = 19.49;

    printf("Το βιβλίο %s εκδόθηκε το %d και κοστίζει %d ευρώ.\n",
           book1.title, book1.year, book1.price);
    return 0;
}
```

Το βιβλίο Programming in C εκδόθηκε το 2015 και κοστίζει 17 ευρώ.

ΕΝΑΛΛΑΚΤΙΚΟΣ ΤΡΟΠΟΣ ΑΡΧΙΚΟΠΟΙΗΣΗΣ ΔΟΜΗΣ

Οι τιμές των πεδίων μιας δομής μπορούν (εναλλακτικά) να αρχικοποιηθούν ταυτόχρονα με τη δήλωση αυτής, π.χ.

```
struct book book1 = {"Programming in C", 2015, 19.49};
```

Εάν αρχικοποιήσουμε μόνο το πρώτο πεδίο, τότε τα υπόλοιπα παίρνουν μηδενική τιμή, π.χ. αν

```
struct book book1 = {"Programming in C", 2015, 19.49};
```

τότε, `book1.year = 0` και `book1.price = 0`.

Μπορούμε κατά την αρχικοποίηση, να δώσουμε τιμή 0 σε όλα τα πεδία ως εξής: `struct book book1 = {0};` Σε αυτή την περίπτωση, επειδή το `book1.title` είναι κείμενο, γίνεται ο χαρακτήρας `'\0'` και όχι ο ακέραιος 0.

ΑΡΧΙΚΟΠΟΙΗΣΗ ΤΑΥΤΟΧΡΟΝΑ ΜΕ ΟΡΙΣΜΟ ΠΡΟΤΥΠΟΥ

Εάν η δήλωση μιας δομής γίνει **ταυτόχρονα με τον ορισμό του προτύπου**, τότε μπορούμε επίσης ταυτόχρονα να αρχικοποιήσουμε και τις τιμές των πεδίων της δομής

π.χ.

```
struct book
{
    char title[100];
    int year;
    float price;
} book1 = {"Programming in C", 2015, 19.49};
```

ΑΝΤΙΓΡΑΦΗ ΔΟΜΩΝ

Με τον τελεστή = μπορούμε να αντιγράψουμε τα πεδία μιας δομής σε μια άλλη του ίδιου προτύπου:

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

struct student
{
    char name[50];
    int regnumber;
    float grade;
};

int main(void)
{
    struct student student1, student2;

    strcpy( student1.name, "Γιώργος Παπαδόπουλος" );
    student1.regnumber = 1234;
    student1.grade = 7.5;

    student2 = student1;

    printf("Ο βαθμός του φοιτητή %s με αριθμό μητρώου %d είναι %2.1f.\n",
           student2.name, student2.regnumber, student2.grade);

    return 0;
}
```

Ο βαθμός του φοιτητή Γιώργος Παπαδόπουλος με αριθμό μητρώου 1234 είναι 7.5.

ΣΥΓΚΡΙΣΗ ΔΟΜΩΝ

Εκτός από την αντιγραφή με $=$, καμία άλλη ενέργεια δεν επιτρέπεται μεταξύ δομών.

Δηλαδή, δεν επιτρέπεται να χρησιμοποιήσουμε τους τελεστές $==$ ή $!=$ για να συγκρίνουμε δυο δομές.

Αντ' αυτού, αν απαιτείται σύγκριση, θα πρέπει να συγκριθούν ένα προς ένα τα μεμονωμένα πεδία που περιέχονται στη δομή.

Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, **Νικόλαος Στεργιούλας**
«Προγραμματισμός Υπολογιστών & Υπολογιστική Φυσική». Έκδοση: 1.0.
Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση:
http://opencourses.auth.gr/eclass_courses.



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Παρόμοια Διανομή [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».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

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





Τέλος ενότητας

Επεξεργασία: Νικόλαος Τρυφωνίδης
Θεσσαλονίκη, 20/09/2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

ΣΗΜΕΙΩΜΑΤΑ

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.

