

Διαδικαστικός Προγραμματισμός

Ενότητα 11: Αρχεία κειμένου

Καθηγήτρια Μαρία Σατρατζέμη

Τμήμα Εφαρμοσμένης Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



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 Πανεπιστήμιο 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



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



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Σκοποί ενότητας

- Να κατανοήσετε την έννοια του *αρχείου κειμένου* και τον τρόπο με τον οποίο αυτά τα αρχεία επιτρέπουν τη μόνιμη αποθήκευση πληροφοριών.
- Να μπορείτε να ανοίγετε και να κλείνετε αρχεία χρησιμοποιώντας τις συναρτήσεις **fopen** και **fclose** αντίστοιχα.
- Να μάθετε να επεξεργάζεστε αρχεία σε κατάσταση χαρακτήρα χρησιμοποιώντας τις συναρτήσεις **getc** και **putc**.
- Να κατανοήσετε την έννοια της μορφοποιημένης εισόδου/εξόδου σε αρχεία.

Η χρησιμότητα των αρχείων

- Κάθε φορά που θέλετε να αποθηκεύσετε πληροφορίες στον υπολογιστή χρονικό διάστημα μεγαλύτερο από το χρόνο εκτέλεσης ενός προγράμματος, μπορείτε να χρησιμοποιείτε ένα **αρχείο** (file).
- Ένα αρχείο είναι μια ομαδοποίηση δεδομένων σε ένα λογικά συνεκτικό σύνολο και η αποθήκευση του σε ένα μέσο μόνιμης αποθήκευσης.

Αρχεία κειμένου

- Όταν αποθηκεύουμε σε ένα αρχείο δεδομένα προκειμένου να χρησιμοποιηθούν από κάποιο πρόγραμμα, αυτό το αρχείο αποτελείται συνήθως από κείμενο και γι' αυτόν το λόγο ονομάζεται **αρχείο κειμένου** (text file).
- Αν υποθέσουμε ότι έχουμε ένα αρχείο, έστω hamlet.txt:

To be or not to be: that is the question.

Whether 'tis nobler in the mind to suffer

αυτό θα αναπαρασταθεί στο εσωτερικό του υπολογιστή ως μια μονοδιάστατη ακολουθία χαρακτήρων

To be or not to be: that is the question.\nWhether 'tis nobler in the mind to suffer \n

- Κάθε αρχείο κειμένου τελειώνει με έναν ειδικό χαρακτήρα τέλους αρχείου που υποδηλώνει ότι δεν υπάρχουν άλλοι χαρακτήρες μετά από εκείνο το σημείο.

Χρήση αρχείων στη C

- Για να χρησιμοποιήσετε κάποιο αρχείο σε ένα πρόγραμμα C, πρέπει να κάνετε τα εξής:
 1. Να δηλώσετε μια μεταβλητή τύπου FILE * (δείκτης αρχείου).
 2. Να συσχετίσετε τη μεταβλητή με ένα πραγματικό αρχείο καλώντας την **fopen**, λειτουργία γνωστή ως **άνοιγμα** (opening) του αρχείου.
 3. Να καλέσετε τις κατάλληλες συναρτήσεις της βιβλιοθήκης **stdio.h** για να εκτελέσετε τις λειτουργίες εισόδου/εξόδου, προκειμένου να διαβάσετε από ή να γράψετε σε αρχεία.
 - Στην περίπτωση ενός αρχείου εισόδου οι συναρτήσεις θα μεταφέρουν δεδομένα από το αρχείο στο πρόγραμμα
 - Στην περίπτωση ενός αρχείου εξόδου οι συναρτήσεις θα μεταφέρουν δεδομένα από το πρόγραμμα στο αρχείο
 4. Να υποδείξετε ότι οι λειτουργίες ολοκληρώθηκαν καλώντας την **fclose**, λειτουργία γνωστή ως **κλείσιμο** (closing) του αρχείου.

Δήλωση μιας μεταβλητής FILE * (1)

- **FILE**: τύπος που ορίζεται στην πρότυπη βιβλιοθήκη εισόδου/εξόδου και χρησιμοποιείται για την αποθήκευση των πληροφοριών που απαιτούνται από το σύστημα για τη διαχείριση της δραστηριότητας των αρχείων.
- Για να μπορέσετε να διαχειριστείτε ένα αρχείο πρέπει να δηλώσετε στη C ένα δείκτη προς μια δομή **FILE** ως εξής:

FILE *infile;

- Ένας **δείκτης** (pointer) είναι ένα στοιχείο δεδομένων του οποίου η τιμή είναι η **διεύθυνση που έχει στη μνήμη κάποια άλλη τιμή**.
- Έτσι, στην περίπτωση μας η μεταβλητή **infile** «δείχνει» προς τη δομή του συστήματος αρχείων.
- Με το ίδιο ακριβώς τρόπο μπορούμε να ορίσουμε μεταβλητές-δείκτες προς μια **int**, **double** κτλ **τιμή**:

int *x;

Δήλωση μιας μεταβλητής FILE * (2)

- Αν σε ένα πρόγραμμα χρειαζόμαστε περισσότερα από ένα αρχεία τότε πρέπει να δηλώσουμε ισάριθμες μεταβλητές τύπου **FILE *** ως εξής:

FILE *infile, *outfile;

- Στη συνέχεια, στις κλήσεις των συναρτήσεων για το διάβασμα από ένα αρχείο ή την εγγραφή σε ένα αρχείο θα χρησιμοποιούμε ως ορίσματα τις παραπάνω μεταβλητές τύπου **FILE ***.

Άνοιγμα αρχείου (1)

Όταν δηλώνετε μια μεταβλητή τύπου **FILE *** για πρώτη φορά, αυτή δεν συσχετίζεται με κάποιο πραγματικό αρχείο. Για να γίνει αυτός ο συσχετισμός πρέπει να καλέσετε τη συνάρτηση **fopen**:

μεταβλητή δείκτη προς αρχείο = fopen(όνομα αρχείου, τρόπος);

Ο τρόπος μεταφοράς δεδομένων είναι ένας από τους εξής:

- **“r”** ανοίγει το αρχείο για **ανάγνωση** (reading). Το αρχείο θα πρέπει να υπάρχει ήδη και η μεταβλητή δείκτη που θα επιστραφεί μπορεί να χρησιμοποιηθεί μόνο σε λειτουργίες εισόδου.
- **“w”** ανοίγει το αρχείο για **εγγραφή** (writing). Αν το αρχείο υπάρχει τότε σβήνονται τα περιεχόμενα του, διαφορετικά δημιουργείται. Η μεταβλητή δείκτη που θα επιστραφεί μπορεί να χρησιμοποιηθεί μόνο σε λειτουργίες εξόδου.
- **“a”** ανοίγει το αρχείο για **προσάρτηση** (appending). Ίδιο με το “w”, αλλά εάν το αρχείο υπάρχει, τα νέα δεδομένα γράφονται στο τέλος των ήδη υπαρχόντων.

Άνοιγμα αρχείου (2)

Παράδειγμα:

- Άνοιγμα του αρχείου **juliet.txt** για είσοδο και συσχέτιση με τη μεταβλητή **infile**:

```
infile = fopen ("juliet.txt", "r");
```

Αν το αρχείο εισόδου λείπει ή ανιχνευθούν άλλα λάθη η κλήση της **fopen** θα επιστρέψει το δείκτη **NULL**.

while (TRUE)

```
{  
    printf ("Input file name: ");  
    gets(filename);  
    infile = fopen (filename, "r");  
    if (infile != NULL) break;  
    printf ("Cannot open file %s. Try again.\n", filename);  
}
```

Εκτέλεση λειτουργιών εισόδου/εξόδου

Αφού ανοίξουμε ένα αρχείο, το επόμενο βήμα είναι να διαβάσουμε ή να γράψουμε τα δεδομένα. Αυτό μπορεί να γίνει επιλέγοντας κάποια από τις παρακάτω στρατηγικές, ανάλογα με την εφαρμογή:

- **Διάβασμα/εγγραφή ανά χαρακτήρα:** χρησιμοποιούνται οι συναρτήσεις `getc()` και `putc()` αντίστοιχα.
- **Διάβασμα/εγγραφή ανά γραμμή:** χρησιμοποιούνται οι συναρτήσεις `fgets()` και `fputs` (διασύνδεση `stdio.h`) αντίστοιχα, ή η `ReadLine()` (αν η διαχείριση των αλφαριθμητικών γίνει με τη βιβλιοθήκη `strlib.h`).
- **Μορφοποιημένη είσοδος και έξοδος:** χρησιμοποιούνται οι συναρτήσεις `fscanf()` και `fprintf()` αντίστοιχα.

Κλείσιμο αρχείων

- Το κλείσιμο ενός αρχείου γίνεται με την κλήση της συνάρτησης **fclose** στην οποία μεταβιβάζεται ο κατάλληλος δείκτης προς αρχείο:
fclose (infile);
fclose (outfile);
- Είναι καλό να κλείνετε τα αρχεία με τη συνάρτηση **fclose** γιατί έτσι:
 - οι αναγνώστες ενός προγράμματος μπορούν εύκολα να καταλαβαίνουν πότε ένα αρχείο είναι σε χρήση και πότε δεν χρειάζεται πλέον.
 - είναι ευκολότερο να ενσωματώνουμε τον κώδικα μας σε μεγαλύτερα προγράμματα που ενδέχεται να διαχειρίζονται δικά τους αρχεία.

Καθιερωμένα αρχεία

- Στη πρότυπη βιβλιοθήκη εισόδου/εξόδου **stdio.h** ορίζονται 3 ειδικά προσδιοριστικά τα **stdin**, **stdout**, **stderr** τα οποία λειτουργούν ως σταθερές τύπου **FILE *** και είναι διαθέσιμα σ' όλα τα προγράμματα. Αυτές οι σταθερές είναι γνωστές ως **καθιερωμένα αρχεία** (standard files).
- Η σταθερά **stdin**, υποδεικνύει το καθιερωμένο αρχείο για την είσοδο δεδομένων από το χρήστη, αναπαριστά τη συσκευή απ' όπου ο χρήστης εισάγει δεδομένα, που είναι το **πληκτρολόγιο**
- Η σταθερά **stdout**, υποδεικνύει το καθιερωμένο αρχείο για την έξοδο δεδομένων και αναπαριστά τη συσκευή στην οποία γράφονται τα δεδομένα εξόδου, που είναι η **οθόνη**
- Η σταθερά **stderr** αναπαριστά το καθιερωμένο αρχείο σφαλμάτων, που είναι η **οθόνη**.
- Σε ορισμένα συστήματα είναι δυνατή η αλλαγή αυτών των καθιερωμένων συσχετίσεων έτσι η καθιερωμένη είσοδος να προέρχεται από αρχείο ή καθιερωμένη έξοδος να κατευθύνεται σε αρχείο ή άλλη συσκευή.

Είσοδος και έξοδος χαρακτήρων (1)

- Η απλούστερη προσέγγιση για την επεξεργασία των αρχείων είναι να διατρέχετε τα αρχεία χαρακτήρα προς χαρακτήρα.

- Χρησιμοποιείτε τη συνάρτηση **getc** ή την ισοδύναμη **fgetc**
int getc(FILE *infile);

Η οποία διαβάζει και επιστρέφει τον επόμενο χαρακτήρα του αρχείου **infile**. Η τιμή που επιστρέφει είναι ένας ακέραιος (και όχι ο χαρακτήρας) έτσι ώστε ο τύπος του αποτελέσματος να μπορεί να περιλαμβάνει και τη ειδική τιμή **EOF** που σηματοδοτεί το τέλος του αρχείου. Η **EOF** έχει συνήθως τη τιμή -1.

- Η συνάρτηση **int getchar()** λειτουργεί με τον ίδιο τρόπο με τη διαφορά ότι διαβάζει πάντα τα δεδομένα εισόδου από τη καθιερωμένη είσοδο το πληκτρολόγιο.

- Για να γράψετε ένα μεμονωμένο χαρακτήρα σε αρχείο χρησιμοποιείτε τη συνάρτηση

int putc(int ch, FILE *outfile);

η οποία γράφει το χαρακτήρα **ch** στο αρχείο εξόδου **outfile**

- Αντίστοιχη είναι η συνάρτηση **int putchar()** η οποία γράφει το χαρακτήρα στη καθιερωμένη έξοδο – οθόνη

Είσοδος και έξοδος χαρακτήρων (2)

```
/* Αντιγραφή ενός αρχείου σε ένα άλλο. */
#include <stdio.h>
#include "genlib.h"

/* Private function prototypes */
static void CopyFile(FILE *infile, FILE *outfile);
static FILE *OpenUserFile(char *prompt, char *mode);

/* Main program */
main()
{
    FILE *infile, *outfile;
    printf("This program copies one file to another.\n");
    infile = OpenUserFile("Old file: ", "r");
    outfile = OpenUserFile("New file: ", "w");
    CopyFile(infile, outfile);
    fclose(infile);
    fclose(outfile);
}
```


Είσοδος και έξοδος χαρακτήρων (3)

```
void CopyFile(FILE *infile, FILE *outfile)
{
    int ch;

    while (TRUE) {
        ch = getc(infile);
        if (ch == EOF) break;
        putc (ch, outfile);
    }
}

FILE *OpenUserFile(char *prompt, char *mode)
{
    char filename[30];
    FILE *result;

    while (TRUE) {
        printf("%s", prompt);
        gets(filename);
        result = fopen(filename, mode);
        if (result != NULL) break;
        printf("Can't open the file \"%s\"\n", filename);
    }
    return (result);
}
```

Συνδυασμός λειτουργιών ανάγνωσης χαρακτήρα και ελέγχου για τη συνθήκη τέλους του αρχείου

```
while ((ch = getc(infile)) != EOF) {
    putc(ch, outfile);
}
```

Είσοδος και έξοδος γραμμών κειμένου

- Επειδή τα αρχεία υποδιαιρούνται συνήθως σε γραμμές, βολεύει να διαβάσετε μια ολόκληρη γραμμή δεδομένων τη φορά. Η συνάρτηση **char *fgets (char buffer[], int bufSize, FILE *infile)** ή **char *fgets (char *buffer, int bufSize, FILE *infile)** διαβάζει την επόμενη γραμμή δεδομένων του αρχείου και την αποθηκεύει στον πίνακα χαρακτήρων **buffer[]** (ή στον πίνακα χαρακτήρων στο οποίο δείχνει ο δείκτης **buffer**). Σταματά το διάβασμα όταν διαβάσει **bufSize-1** χαρακτήρες, ή το χαρακτήρα αλλαγής γραμμής, ή το χαρακτήρα τέλους του αρχείου, οποιοδήποτε συμβεί πρώτο.
- Προσθέτει το χαρακτήρα τερματισμού ('\0') στους χαρακτήρες που διαβάστηκαν.
- Αν η συνάρτηση εκτελεστεί επιτυχώς τότε επιστρέφει ως τιμή ένα δείκτη στον πίνακα χαρακτήρων **buffer** αλλιώς τη τιμή **NULL**.
- Η αντίστοιχη συνάρτηση για την έξοδο δεδομένων είναι η **int fputs(char *buffer, FILE *outfile)** αντιγράφει το αλφαριθμητικό **buffer** στο αρχείο εξόδου **outfile** μέχρι να εμφανισθεί ο χαρακτήρας τέλους του αλφαριθμητικού

Μορφοποιημένη είσοδος και έξοδος (1)

Η συνάρτηση **printf** έχει τρεις διαφορετικές μορφές:

- **printf (αλφαριθμητικό ελέγχου, ...);**
στέλνει τα δεδομένα εξόδου στην καθιερωμένη έξοδο (οθόνη ΗΥ)
- **fprintf (ρεύμα εξόδου, αλφαριθμητικό ελέγχου, ...);**
στέλνει τα δεδομένα εξόδου σε αρχείο
- **sprintf (πίνακας χαρακτήρων, αλφαριθμητικό ελέγχου, ...);**
στέλνει τα δεδομένα εξόδου σε πίνακα

Μορφοποιημένη είσοδος και έξοδος (2)

Η συνάρτηση *scanf* έχει τρεις διαφορετικές μορφές:

- ***int scanf (αλφαριθμητικό ελέγχου, ...);***
Διαβάζει τα δεδομένα εισόδου από την καθιερωμένη είσοδο
- ***int fscanf (ρεύμα εισόδου, αλφαριθμητικό ελέγχου, ...);***
Διαβάζει τα δεδομένα εισόδου από αρχείο (το δείκτη FILE που καθορίζεται με την παράμετρο ρεύμα εισόδου)
- ***sscanf (αλφαριθμητικό χαρακτήρων, αλφαριθμητικό ελέγχου, ...);***
Διαβάζει τα δεδομένα εισόδου από το καθορισμένο αλφαριθμητικό χαρακτήρων
- Επειδή η *scanf* πρέπει να επιστρέφει πληροφορίες σ' αυτόν που την καλεί γι' αυτό τα ορίσματα που ακολουθούν το αλφαριθμητικό έλεγχο πρέπει να καθορίζονται με κλήση κατ' αναφορά (δείκτες) (εκτός από πίνακες και πίνακες χαρακτήρων που αποτελούν δείκτες από μόνα τους).

Παράδειγμα (1)

```
#include <stdio.h>
#include "genlib.h"
#include "simpio.h"

#define ElementsFile "elements.dat"
#define MaxElementName 15
#define MaxSymbolName 2

main()
{
    FILE *infile;
    char elementName[MaxElementName+1];
    char elementSymbol[MaxSymbolName+1];
    char namebuf[MaxElementName+MaxSymbolName+4];
    int atomicNumber;
    double atomicWeight;
    char termch;
    int nscan;
```

Παράδειγμα (2)

```
infile = fopen(ElementsFile, "r");

if (infile == NULL) Error("Can't open %s", ElementsFile);

printf("  Element (symbol)  Atomic Weight\n");
printf("-----\n");

while (TRUE) {
  nscan = fscanf(infile, "%15[^,], %2[^,], %d, %lf%c",
                elementName, elementSymbol,
                &atomicNumber, &atomicWeight,
                &termch);
  if (nscan == EOF) break;
  if (nscan != 5 || termch != '\n') {
    Error("Improper file format");
  }
  sprintf(namebuf, "%s (%s)", elementName, elementSymbol);
  printf("%3d, %-20s %8.3f\n", atomicNumber, namebuf,
        atomicWeight);
}
}
```

Δείτε το αρχείο Table15-1.pdf για προδιαγραφές μετατροπής της συνάρτησης scanf

Διορθώσεις στο βιβλίο σχετικά με τα αρχεία

- Στη σελίδα 620 αντικαταστήστε το λανθασμένη πρωτότυπο της συνάρτησης `fputs` από
`void fputs(string str, FILE *outfile);`
σε
`int fputs(string str, FILE *outfile);`
- Στη σελίδα 769 αντικαταστήστε το λανθασμένη πρώτυπο της συνάρτησης `putc` από
`void putc(char ch, FILE *outfile);`
σε
`int putc(int ch, FILE *outfile);`
- Σ'όλο το κεφάλαιο 15 αλλά και στο παράρτημα A.9 (σελίδες 759-762 βιβλιοθήκη `stdio.h`) **αντικαταστήστε** στα πρωτότυπα των συναρτήσεων για τη διαχείριση αρχείων, το τύπο **string** (δεν υποστηρίζεται από την ANSI C) με το τύπο **char*** ή **char[]**.

Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

