



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

ΑΡΧΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Κεφάλαιο 13

Επιμέλεια:

Βασίλης Παλιουράς , Αναπληρωτής Καθηγητής
Ευάγγελος Δερματάς , Αναπληρωτής Καθηγητής
Σταύρος Νούσιος , Βοηθός Ερευνητή

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών

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.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου των διδασκόντων καθηγητών.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών» έχει χρηματοδοτηθεί 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.



Ανάπτυξη

Το παρόν εκπαιδευτικό υλικό αναπτύχθηκε στο τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Stack

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

```
int f (int );  
int g (int );  
double h (double );  
int w (int, int);
```

Που αποθηκεύονται
παράμετροι και τοπικές μεταβλητές:
stack

```
int main(int argc, char *argv[]) {
```

```
    f(1); ← H f δεν καλεί την g
```

```
    g(1);
```

```
    h(1.0);
```

```
    w(1, 2);
```

```
    f(2);
```

```
    g(1);
```

```
    return 0;
```

```
}
```

 **H f καλεί την g**

```
int f(int a) {  
    int b = 1 ;  
    printf("function f: address of parameter %X\n", &a);  
    printf("\t\t address of local variable %X\n", &b);  
    if (a>1)  
        g(a);  
    return b + a;  
}
```

Διεύθυνση στη μνήμη

```
int g (int a) {  
    int b = 1 ;  
    int *c = &a;  
    printf("function g: address of parameter %X\n", &a);  
    printf("\t\t address of local variable b %X\n", &b);  
    printf("\t\t address of local variable c %X\n", &c);  
    return b + a ;  
}
```

```
int w (int a, int c) {  
    int b = 1 ;  
    printf("function w: address of parameter %X\n", &a);  
    printf("\t\t address of local variable %X\n", &b);  
    return b + a + c;  
}
```

```
double h (double a) {  
    double b = 2.0 ;  
    printf("function h: address of parameter %X\n", &a);  
    printf("\t\t address of local variable %X\n", &b);  
    return b + a;  
}
```



Stack

Εικόνα του stack κατά τη διάρκεια εκτέλεσης των συναρτήσεων

Κάθε φορά διατίθεται ο ίδιος χώρος μνήμης

```

C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
function f: address of parameter 22FE40
              address of local variable 22FE2C
function g: address of parameter 22FE40
              address of local variable b 22FE2C
              address of local variable c 22FE20
function h: address of parameter 22FE40
              address of local variable 22FE28
function w: address of parameter 22FE40
              address of local variable 22FE2C
function f: address of parameter 22FE40
              address of local variable 22FE2C
function g: address of parameter 22FE00
              address of local variable b 22FDEC
              address of local variable c 22FDE0
function g: address of parameter 22FE40
              address of local variable b 22FE2C
              address of local variable c 22FE20

-----
Process exited with return value 0
Press any key to continue . . .
    
```

Χώρος f που κ αλείται από main	Χώρος h που κ αλείται από main	Χώρος g που κ αλείται από main	Χώρος g που κ αλείται από main	Χώρος f που κ αλείται από main	Χώρος w που καλείται από main
				Χώρος g που κ αλείται από f	

Αναγκαστικά διαφορετικός γιατί η f καλεί τη g



Βασική διαχείριση αρχείων

- Τύπος FILE
- Μερικές συναρτήσεις
 - fopen()
 - fclose()
 - fprintf()
 - fscanf()
 - feof()
 - ...



fopen() - fclose()

```
#include <stdio.h>
#define N 10
```

```
int main(int argc, char *argv[]) {
```

```
FILE *dataout;
int i;
```

← Δείκτης σε FILE

```
dataout = fopen ("mydata.txt", "wt");
```

↙ Αρχείο για εγγραφή κειμένου
(ascii)

```
for (i =0 ;i < N; i++)
    fprintf(dataout, "%d\n", i);
```

```
fclose(dataout);
```

```
return 0;
```



Διαβάζω έναν αριθμό

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
```

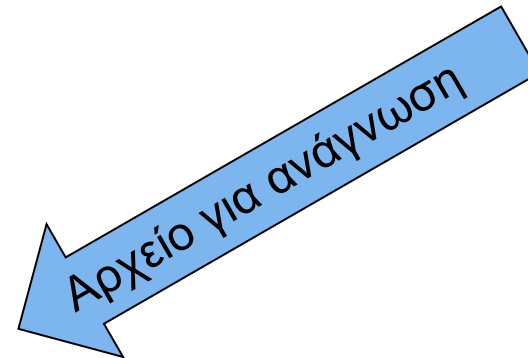
```
    FILE *myinput;
    int d = 0;
```

```
    if ((myinput = fopen("mydata2.txt", "rt"))==NULL)
    {
        printf("cannot open file\n");
        exit(1);
    }
```

```
    fscanf(myinput, "%d", &d);
    printf("%d", d);
```

```
    fclose(myinput);
    return 0;
```

```
}
```



Αν δεν μπορέσει να ανοίξει το αρχείο (πχ δεν υπάρχει) τότε το myinput λαμβάνει την τιμή 0 (NULL)



Διαβάζω N αριθμούς

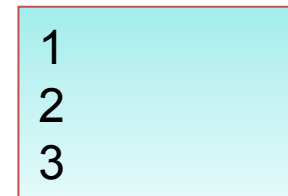
```
#include <stdio.h>
#include <stdlib.h>
#define N 3
int main(int argc, char *argv[]) {
    FILE *myinput;
    int d, i, j;

    myinput = fopen("mydata2.txt", "rt");
    if (myinput == NULL)
    {
        printf("cannot read file\n");
        exit(1);
    }

    for (i = 0; i < N ; i++) {
        fscanf(myinput, "%d", &d);
        printf("%d\n", d);
    }

    fclose(myinput);

    return 0;
}
```



1
2
3

Περιεχόμενα αρχείου
mydata2.txt



Η συνάρτηση fscanf()

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

Επιστρέφει το πλήθος των ορισμάτων που διαβάστηκαν επιτυχώς.

Επιστρεφόμενη τιμή: Ακέραιος ίσος ή μικρότερος από το πλήθος των ορισμάτων στα οποία καλείται να δώσει τιμή.

Σε περίπτωση EOF, (end-of-file) πριν διαβαστεί στοιχείο, επιστρέφει EOF.



Περισσότερες αναγνώσεις από αριθμούς που υπάρχουν στο αρχείο

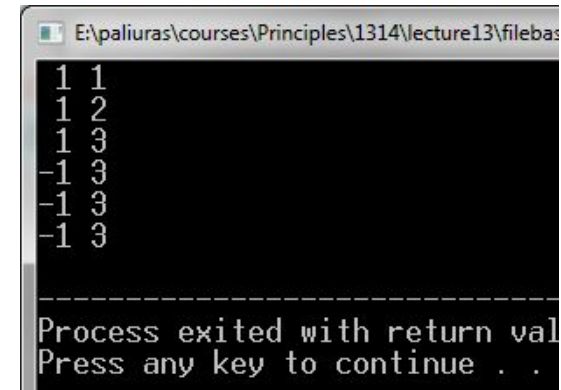
```
#include <stdio.h>
#include <stdlib.h>
#define N 3
int main(int argc, char *argv[]) {
    FILE *myinput;
    int d, i, j, a;

    myinput = fopen("mydata2.txt", "rt");
    if (myinput == NULL)
    {
        printf("cannot read file\n");
        exit(1);
    }

    for (i = 0; i < 2*N ; i++) {
        a = fscanf(myinput, "%d", &d);
        printf("%2d %d\n", a, d);
    }

    fclose(myinput);

    return 0;
}
```



```
E:\paliuras\courses\Principles\1314\lecture13\filebas
1 1
1 2
1 3
-1 3
-1 3
-1 3
-----
Process exited with return val
Press any key to continue . .
```

a, γίνεται -1 (EOF) όταν δεν μπορέσει να διαβάσει γιατί έφτασε στο τέλος του αρχείου. Τότε το d δεν παίρνει νέα τιμή.



Παράδειγμα 1

```
#include <stdio.h>
#include <stdlib.h>
#define N 3
int main(int argc, char *argv[]) {
    FILE *myinput;
    int d, i, j, a;

    if ((myinput = fopen("mydata2.txt", "rt"))== NULL)
    {
        printf("cannot read file\n");
        exit(1);
    }

    for (a = 0; a > -1 ; ) {
        a = fscanf(myinput, "%d", &d);
        printf("%2d %d\n", a, d);
    }

    fclose(myinput);

    return 0;
}
```



```
E:\paliuras\courses\Principles\1314\lecture13\file
1 1
1 2
1 3
-1 3
-----
Process exited with return v
Press any key to continue
```

Παράδειγμα 2

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

int main(int argc, char *argv[]) {
    FILE *myinput;
    int d, a;

    if ((myinput = fopen("mydata2.txt", "rt"))== NULL)
    {
        printf("cannot read file\n");
        exit(1);
    }

    for (a = 0; a > -1 ; ) {
        a = fscanf(myinput, "%d", &d);
        if (a>-1)
            printf("%2d %d\n", a, d);
    }

    fclose(myinput);

    return 0;
}
```



```
E:\paliuras\courses\Principles\1314\lecture13\filebasicInNu
1 1
1 2
1 3
-----
Process exited with return value 0
```

Παράδειγμα 3

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

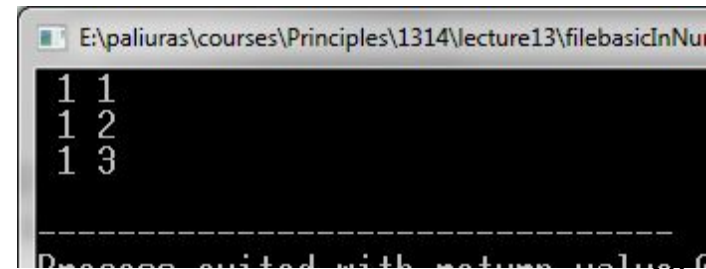
int main(int argc, char *argv[]) {
    FILE *myinput;
    int d, a;

    if ((myinput = fopen("mydata2.txt", "rt"))== NULL)
    {
        printf("cannot read file\n");
        exit(1);
    }

    for (; (a = fscanf(myinput, "%d", &d)) > -1 ; ) {
        printf("%2d %d\n", a, d);
    }

    fclose(myinput);

    return 0;
}
```



```
E:\paliuras\courses\Principles\1314\lecture13\filebasicInNu
1 1
1 2
1 3
-----
Process exited with return value 0
```


Παράδειγμα 4

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

int main(int argc, char *argv[]) {
    FILE *myinput;
    int d, a;

    if ((myinput = fopen("mydata2.txt", "rt"))== NULL)
    {
        printf("cannot read file\n");
        exit(1);
    }

    for (; (a = fscanf(myinput, "%d", &d)) != EOF ; )
        printf("%2d %d\n", a, d);

    fclose(myinput);

    return 0;
}
```



Παράδειγμα 5

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

int main(int argc, char *argv[]) {
    FILE *myinput;
    int d, a;

    if ((myinput = fopen("mydata2.txt", "rt"))== NULL)
    {
        printf("cannot read file\n");
        exit(1);
    }

    while ((a = fscanf(myinput, "%d", &d)) != EOF )
        printf("%2d %d\n", a, d);
    fclose(myinput);
    return 0;
}
```



Παράδειγμα 6

Δημιουργία αρχείου για εγγραφή

```
#include <stdio.h>

void createfile (void);
void replaceinfile (void);

int main(int argc, char *argv[]) {
    FILE *in;

    createfile();

    printf("file created\n");
    system("pause");

    replaceinfile();
    printf("file updated\n");

    return 0;
}
```

```
void createfile(void) {
    FILE *a;
    a = fopen("mytest.txt", "wt");
    fprintf(a, "my message: hello there!\n");
    fclose(a);
}

void replaceinfile() {
    FILE *a;
    a = fopen("mytest.txt", "r+");
    fseek(a, 12, SEEK_SET);
    fprintf(a, "goodbye all");
    fclose(a);
}
```



Πήγαινε 12 θέσεις από την Αρχή του αρχείου

Άνοιγμα υπάρχοντος αρχείου για ανάγνωση/ενημέρωση

Παράδειγμα 7

```
#include <stdio.h>
#define N 10

int main(int argc, char *argv[]) {

    FILE *dataout;
    int i;

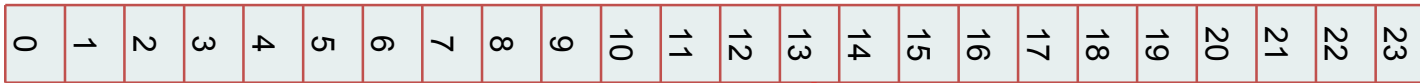
    dataout = fopen ("mydata.txt", "wt");

    for (i =0 ;i < N;  i++)
        fprintf(dataout, "%d\n", i);

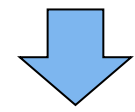
    fclose(dataout);

    return 0;
}
```





m y m e s s a g e : h e l l o t h e r e !



m y m e s s a g e : g o o d b y e a l l !



long int ftell(FILE *)

```
#include <stdio.h>
void createfile (FILE *);
void replaceinfile (FILE *);
long int getfilesize(FILE *);

int main(int argc, char *argv[]) {
    FILE *in;
    long int length;
    createfile(in);
    printf("file created\n");
    system("pause");
    replaceinfile(in);
    printf("file updated\n");
    length = getfilesize(in);
    printf("Contains: %ld chars\n",
           length);

    return 0;
}
```

```
void replaceinfile(FILE *a) {
    a = fopen("mytest.txt", "r+");
    fseek(a, 12, SEEK_SET);
    printf("location: %ld\n", ftell(a));
    fprintf(a,"goodbye all");
    printf("location: %ld\n", ftell(a));
    fclose(a);
}
```

```
long int getfilesize(FILE *a) {
    long int size;
    a = fopen("mytest.txt", "rt");
    fseek(a, 0, SEEK_END);
    size = ftell(a);
    fclose(a);
    return size;
}
```



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



Απόσταση από σημείο αναφοράς



Ορισμός σημείου αναφοράς

Σταθερές ορισμού σημείου αναφοράς	Σημείο αναφοράς
SEEK_SET	Αρχή αρχείου
SEEK_CUR	Τρέχουσα θέση στο αρχείο
SEEK_END	Τέλος αρχείου



Παραδειγμα 8

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

```
void dosomething(FILE *);  
int main(int argc, char *argv[]) {  
    FILE *p;
```

```
    p = fopen("myfile.txt", "wt");
```

```
    dosomething(p);
```

```
    fclose(p);
```

```
    return 0;
```

```
}
```

```
void dosomething(FILE *f) {  
    fprintf(f, "a message.");
```

Το αρχείο ανοίγει
Στη συνάρτηση που
Καλεί τη συνάρτηση
Που κάνει επεξεργασία



Βρείτε τις διαφορές

```
#include <stdio.h>
void dosomething(int );
int main(int argc, char *argv[]) {
    int a ;

    a = 5;
    dosomething(a) ;
    printf("a main: %d\n", a);

    return 0;
}

void dosomething(int a) {
    printf("a function: %d\n", a);
    a ++;
    printf("a updated function: %d\n", a);
}
```

```
#include <stdio.h>
void dosomething(int *);
int main(int argc, char *argv[]) {
    int a ;

    a = 5;
    dosomething(&a) ;
    printf("a main: %d\n", a);

    return 0;
}

void dosomething(int *a) {
    printf("a function: %d\n", *a);
    (*a) ++;
    printf("a updated function: %d\n", *a);
}
```



Παραδειγμα 9

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

```
void fileopen(FILE **);  
void fileclose(FILE *);  
void filecomplete(FILE *);
```

```
int main(int argc, char *argv[]) {  
    FILE *p;  
  
    fileopen(&p);  
  
    filecomplete(p);  
  
    fileclose(p);  
  
    return 0;
```

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

```
void fileopen(FILE **p) {  
    *p = fopen("test.txt", "wt");  
  
}
```



Παραδειγμα 10

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

```
FILE * fopen(void);  
void fclose(FILE *);  
void filecomplete(FILE *);
```

```
int main(int argc, char *argv[]) {  
    FILE *p;  
  
    p = fopen();  
  
    filecomplete(p);  
  
    fclose(p);  
  
    return 0;  
}
```

```
FILE * fopen(void) {  
    FILE *p;  
  
    p = fopen("test.txt","wt");  
    return p;  
}
```



Παραδειγμα 11

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

```
void filecomplete(FILE *p) {  
    int i;  
    for (i = 0 ; i < 10 ; i++) {  
        fprintf(p, "hello %d\n", i);  
    }  
}
```

```
#include <stdio.h>  
void fileclose(FILE *p) {  
    fclose(p);  
}
```



Σημείωμα αναφοράς

- Copyright Πανεπιστήμιο Πατρών,
Παλιουράς Βασίλειος , Δερματάς Ευάγγελος
«Αρχές Προγραμματισμού ».
Έκδοση: 1.0. Πάτρα 2015
- Διαθέσιμο από τη δικτυακική διεύθυνση
<https://eclass.upatras.gr/modules/>

