

# ΔΙΑΔΙΚΑΣΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

7<sup>η</sup> Εβδομάδα: Διαχείριση Αρχείων

# Αναφορές

Οι διαφάνειες της διάλεξης στηρίζονται, εν μέρει, σε υλικό παραδόσεων παλαιότερων ετών του **Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογία Υπολογιστών του Πανεπιστημίου Πατρών** καθώς και του **Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου**.

# Πρόσβαση σε Αρχεία

- Κάθε αρχείο είναι μια συλλογή από δεδομένα αποθηκευμένα στη δευτερεύουσα μνήμη.
- Για πρόσβαση σε ένα αρχείο από το πρόγραμμα μας, πρέπει πρώτα να μεταφερθεί στην Κύρια Μνήμη και να δημιουργηθεί **σύνδεση μεταξύ αρχείου και προγράμματος!**

# FILE: Μεταβλητή Σύνδεσης με Αρχείο

- Μέσα στο πρόγραμμα μας, η σύνδεση με το αρχείο παίρνει τη μορφή μιας μεταβλητής.
- Ο τύπος της μεταβλητής είναι γνωστός και ως **filestream (FILE)** που σημαίνει σύνδεση ροής με αρχείο.
- Πιο συγκεκριμένα η μεταβλητή αυτή θα περιέχει ένα δείκτη ο οποίος θα δείχνει σε ένα συγκεκριμένο σημείο του αρχείου.

# Σύνδεση με Αρχείο **fopen()**

- Για να δημιουργήσουμε την σύνδεση με το αρχείο χρησιμοποιούμε τη συνάρτηση **fopen()**, και καθορίζουμε:
  - Το **όνομα του αρχείου** με το οποίο θέλουμε να συνδεθούμε
  - Τον **τρόπο με τον οποίο θέλουμε να συνδεθούμε** (π.χ., αν θέλουμε να το ανοίξουμε για διάβασμα για έγγραφο, ή για προσάρτηση κτλ).

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

# Σύνδεση με Αρχείο **fopen()**

```
FILE *Variable;  
Variable = fopen(filepath, type_of_process)
```

Όνομα συνάρτησης

**1<sup>η</sup> παράμετρος**

**Διαδρομή** (σχετική ή απόλυτη) για το αρχείο που θέλουμε να διαχειριστούμε.

**2<sup>η</sup> παράμετρος**

Τρόπος διαχείρισης του αρχείου:

- διάβασε μόνο
- γράψε μόνο
- Προσάρτησε
- δυαδική ανάγνωση
- δυαδική εγγραφή

# Επιλογές χρήσης (“ανοίγματος”) αρχείου

6

Access Type	Description
r	Opens an existing text file for reading purpose.
w	Opens a text file for writing. If it does not exist, then a new file is created. Here your program will start writing content from the beginning of the file.
a	Opens a text file for writing in appending mode. If it does not exist, then a new file is created. Here your program will start appending content in the existing file content.
r+	Opens a text file for both reading and writing.
w+	Opens a text file for both reading and writing. It first truncates the file to zero length if it exists, otherwise creates a file if it does not exist.
a+	Opens a text file for both reading and writing. It creates the file if it does not exist. The reading will start from the beginning but writing can only be appended.
t  b	text or binary mode

π.χ. Ποια η διαφορά μεταξύ **”wt”** και **”wb”** ;

# Σύνδεση με Αρχείο `fopen()`

- Όταν ανοίγουμε ένα αρχείο για **διάβασμα ("r")**
  - Πρέπει να είμαστε σίγουροι ότι το **αρχείο αυτό υπάρχει!**
  - Μόλις δημιουργηθεί η σύνδεση, ο δείκτης δείχνει στην **αρχή του αρχείου**



# Σύνδεση με Αρχείο `fopen()`

- Όταν ανοίγουμε ένα αρχείο για **γράψιμό** ("`w`"):
  - Αν το αρχείο αυτό υπάρχει τότε **ότι υπάρχει μέσα θα διαγραφεί**.
  - Αν το αρχείο δεν υπάρχει τότε **θα δημιουργηθεί**.
  - Μόλις δημιουργηθεί η σύνδεση, ο δείκτης δείχνει στην **αρχή του αρχείου**.
- Όταν ανοίγουμε ένα αρχείο για **προσάρτηση** ("`a`"):
  - Αν το αρχείο αυτό υπάρχει τότε **το περιεχόμενο του παραμένει ως έχει** και τα καινούρια δεδομένα **καταχωρούνται στο τέλος του αρχείου**.
  - Αν το αρχείο δεν υπάρχει τότε **θα δημιουργηθεί**.
  - Μόλις δημιουργηθεί η σύνδεση, ο δείκτης δείχνει στο **τέλος του αρχείου**.

# Σύνδεση με Αρχείο `fopen()`

## Προσοχή:

- Χρήση του `"w"` πρωτίστως σβήνει όλο το περιεχόμενο του αρχείου!
- Χρήση του `"a"` επιτρέπει την προσάρτιση κειμένου στο τέλος του αρχείου.

# Αποσύνδεση από Αρχείο `fclose()`

- Η αποσύνδεση από το αρχείο γίνεται με τη μέθοδο `fclose()`.

- `fclose(δείκτης στο αρχείο)`

`fclose(readFile);`

`fclose(writeFile);`

- Κατά τη διάρκεια της επεξεργασίας, το αρχείο βρίσκεται στην Κύρια Μνήμη. Η συνάρτηση `fclose()` μεταφέρει το επεξεργασμένο αρχείο στην Δευτερεύουσα μνήμη (για μόνιμη αποθήκευση) και αποδεσμεύει τη μνήμη που κρατούσε η μεταβλητή `readFile` και `writeFile`.
- Αν δεν κλείσουμε τα αρχεία πριν τον τερματισμό του προγράμματος, τότε τα αρχεία κλείνουν ταυτόχρονα με το τέλος της `main()` ή με την εκτέλεση της `return 0` της `main()`

# Εντολή **fprintf**

- Η εντολή **fprintf** είναι ανάλογη της printf αλλά για αρχεία (αντί οθόνης)

π.χ `fprintf(writeFile, "Ο αριθμος είναι: %d", num);`

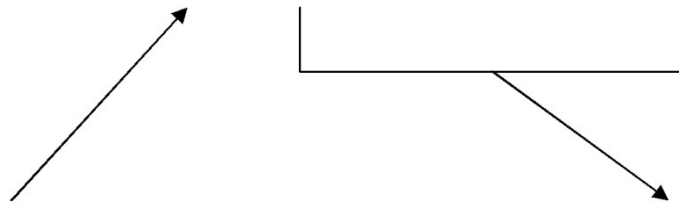
**Δείκτης** στο αρχείο που θα γράψουμε **Όπως η printf**



# Εντολή **fscanf**

- Η εντολή **fscanf** είναι ανάλογη της `scanf` αλλά για αρχεία (αντί πληκτρολογίου)

π.χ `fscanf(readFile, "%d", &num )`



**Δείκτης** στο αρχείο που θα διαβάσουμε **Όπως η `scanf`**

# Βασική διαχείριση αρχείου

- ❖ Δηλώνουμε μεταβλητή τύπου FILE \* για να χειριστούμε το αρχείο
- ❖ Ανοίγουμε το αρχείο με κατάλληλο fopen ( ), όνομα αρχείου τρόπος χρήσης:
  - ❖ ανάγνωση κειμένου,
  - ❖ εγγραφή κειμένου,
  - ❖ δυαδική ανάγνωση,
  - ❖ δυαδική εγγραφή, ...
- ❖ Επεξεργαζόμαστε με κατάλληλες συναρτήσεις
- ❖ Κλείνουμε το αρχείο με κατάλληλο fclose()

# Παράδειγμα

Να γραφεί πρόγραμμα το οποίο να γράφει στο αρχείο **mydata.txt** σε μορφή ASCII τους 10 πρώτους ακεραίους .

Γράφουμε 10  
ακεραίους σε  
αρχείο ASCII

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

```
int main( void ) {
```

```
FILE *dataout;
```

```
int i;
```

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

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

```
fclose(dataout);
```

```
return EXIT_SUCCESS;
```

ΔΕΙΚΤΗΣ σε FILE

Όνομα  
αρχείου

Άνοιγμα αρχείου για  
εγγραφή κειμένου  
(ascii)



# Διαβάζω έναν αριθμό

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

```
int main(void) {
```

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

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

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

```
    fclose(myinput);
```

```
    return EXIT_SUCCESS;
```

```
}
```

Όνομα αρχείου

Άνοιγμα αρχείου για ανάγνωση κειμένου

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

```
#include <stdio.h>
#include <stdlib.h>
#define N 3
```

```
int main(void) {
    FILE *myinput;
    int d, i;

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

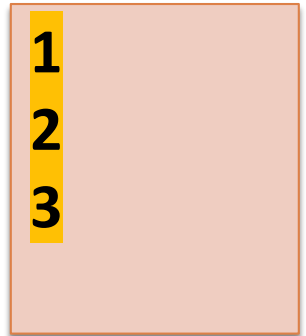
```
for (i = 0; i < N ; i++) {
    fscanf(myinput, "%d", &d);
    printf("%d\n", d);
}
```

```
fclose(myinput);

return EXIT_SUCCESS;
```

# Ανάγνωση γνωστού πλήθους αριθμών

Περιεχόμενα αρχείου  
mydata2.txt



```
}
```

Περισσότερες αναγνώσεις από αριθμούς που υπάρχουν στο αρχείο

```
#include <stdio.h>
#include <stdlib.h>
#define N 3
```

```
int main(void) {
    FILE *myinput;
    int d, i, a;

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

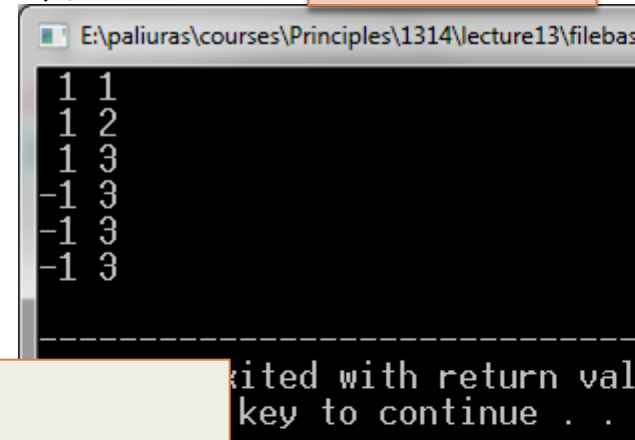
    for (i = 0; i < 2*N ; i++) {
        a = fscanf(myinput, "%d", &d);
        printf("%2d %d\n", a, d);
    }

    fclose(myinput);

    return EXIT_SUCCESS;
}
```

Περιεχόμενα αρχείου  
mydata2.txt

1  
2  
3



```
E:\paliuras\courses\Principles\1314\lecture13\filebas
1 1
1 2
1 3
-1 3
-1 3
-1 3
...
kited with return val
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.

```
#include <stdio.h>
#include <stdlib.h>
#define N 3
```

# Λάθος χρήση του EOF

19

```
int main(void) {
    FILE *myinput;
    int d, a;
```

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

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

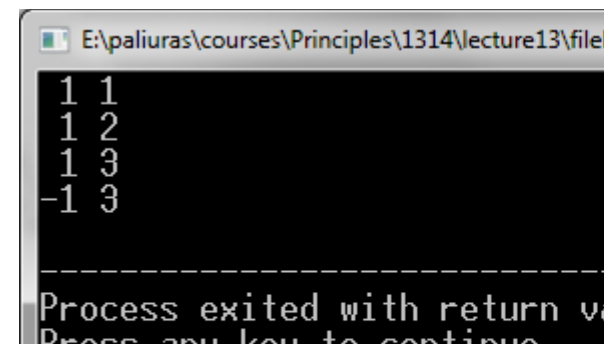
```
    fclose(myinput);

    return EXIT_SUCCESS;
```

```
}
```

Περιεχόμενα αρχείου  
mydata2.txt

1  
2  
3



```
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
```

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

# Διόρθωση

```
int main(void) {
    FILE *myinput;
    int d, a;

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

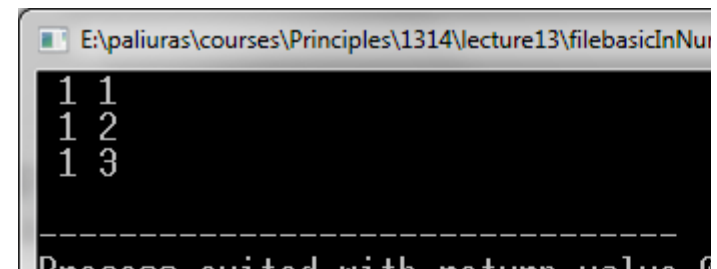
Περιεχόμενα αρχείου  
mydata2.txt

1  
2  
3

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

```
    fclose(myinput);
    return EXIT_SUCCESS;
```

```
}
```



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

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

```
int main(void) {
    FILE *myinput;
    int d, a;

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

Περιεχόμενα αρχείου  
mydata2.txt

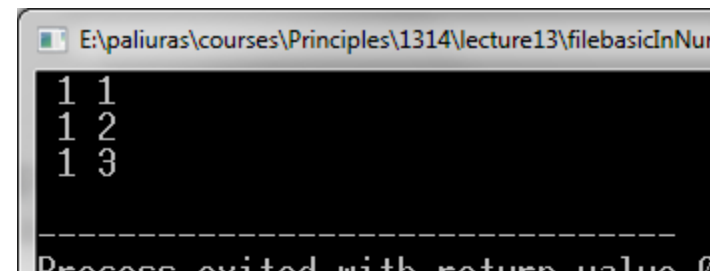
1  
2  
3

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

```
fclose(myinput);

return EXIT_SUCCESS;
```

```
}
```



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

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

```
int main(void) {
    FILE *myinput;
    int d, a;

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

Περιεχόμενα αρχείου  
mydata2.txt

1  
2  
3

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

```
    fclose(myinput);
```

```
    return EXIT_SUCCESS;
```

```
}
```

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

```
int main(void) {
    FILE *myinput;
    int d, a;

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

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

    fclose(myinput);

    return EXIT_SUCCESS;
}
```

Περιεχόμενα αρχείου  
mydata2.txt

1  
2  
3



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

```
int main(void) {
    FILE *myinput;
    int d;

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

    while (fscanf(myinput, "%d", &d)!= EOF ) {
        printf("%2d\n", d);
    }

    fclose(myinput);

    return EXIT_SUCCESS;
}
```

Περιεχόμενα αρχείου  
mydata2.txt

1  
2  
3

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

# Διαβάζω χαρακτήρες

```
int main(void) {

    FILE * textin;
    int ch;

    if ((textin = fopen("mytext.txt", "rt"))==NULL) {
        printf("Cannot open file.\n");
        exit(EXIT_FAILURE);
    }

    while ( (ch = fgetc(textin))!=EOF ) {
        printf("%c", ch);
    }

    fclose(textin);

    return EXIT_SUCCESS;
}
```

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

# Διαβάζω λέξεις

```
int main(void) {

    FILE * textin ;
    char word[100];

    if ( (textin = fopen("mytext.txt", "rt"))==NULL) {
        printf("cannot open file.\n");
        exit(EXIT_FAILURE);
    }

    while (fscanf(textin, "%s", word) != EOF) {
        printf("%s\n", word);
    }

    fclose(textin);

    return EXIT_SUCCESS;
}
```

# Παράδειγμα

● ● ● my\_friends.txt  
XRHSTOS 35 2610345698  
DHMHTRHS 25 2610345695  
GIORGOS 45 2610345692

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

int main()
{
    FILE *readfile;
    int age;
    char phone[20];
    char name[25];
    readfile = fopen ("my_friends.txt", "r");

    if(!readfile){
        printf("Error opening the file");
        return 0;
    }

    while(!feof(readfile)){
        if(fscanf(readfile,"%s %d %s",name,&age,phone)){
            printf("Name: %s, Age:%d, Phone: %s \n", name,age,phone);
        }
    }

    fclose(readfile);
    return 1;
}
```

Να γραφεί πρόγραμμα το οποίο να διαβάζει από το αρχείο my\_friends.txt και να τυπώνει στην οθόνη για όλους τους φίλους που είναι καταχωρημένοι στο αρχείο το όνομα, την ηλικία και το τηλέφωνο τους.

```
Name: XRHSTOS, Age:35, Phone: 2610345698
Name: DHMHTRHS, Age:25, Phone: 2610345695
Name: GIORGOS, Age:45, Phone: 2610345692
```

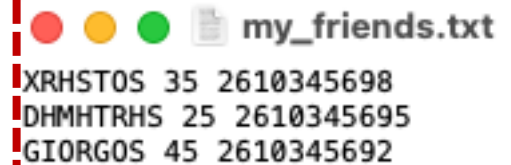
# Παράδειγμα

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

int main()
{
    FILE *readfile;
    int age;
    char phone[20];
    char name[25];
    readfile = fopen ("my_friends.txt", "r");

    if(!readfile){
        printf("Error opening the file");
        return 0;
    }

    while( fscanf(readfile, "%s %d %s", name, &age, phone) != EOF ){
        printf("Name: %s, Age: %d, Phone: %s \n", name, age, phone);
    }
    fclose(readfile);
    return 1;
}
```



my\_friends.txt

```
XRHSTOS 35 2610345698
DHMHTRHS 25 2610345695
GIORGOS 45 2610345692
```

```
Name: XRHSTOS, Age: 35, Phone: 2610345698
Name: DHMHTRHS, Age: 25, Phone: 2610345695
Name: GIORGOS, Age: 45, Phone: 2610345692
```

# Παράδειγμα

```
#include <stdio.h>
#include <stdlib.h>
#define MAXCHAR 1000
int main()
{
    FILE *readfile;
    char buffer[MAXCHAR];

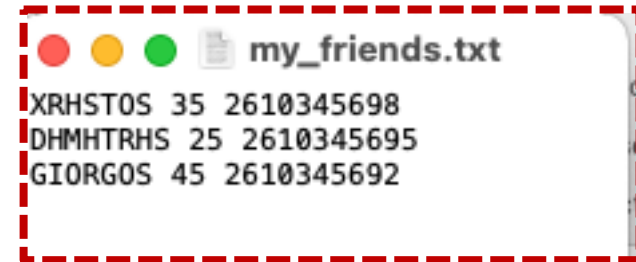
    readfile = fopen ("my_friends.txt", "r");

    if(!readfile){
        printf("Error opening the file");
        return 0;
    }

    while( fgets(buffer,MAXCHAR,readfile)!=NULL ){
        printf("Line: %s", buffer);
    }

    fclose(readfile);
    return 1;
}
```

**char \*fgets(char \*str, int n, FILE \*stream)**  
fgets σταματά την ανάγνωση χαρακτήρων είτε μέχρι να διαβάσει (**MAXCHAR -1**) χαρακτήρες, είτε μέχρι να διαβάσει το *newline character*, είτε μέχρι το *end-of-file*, whichever comes first.



```
Line: XRHSTOS 35 2610345698
Line: DHMHTRHS 25 2610345695
Line: GIORGOS 45 2610345692
```

# Παράδειγμα

```
my_friends.txt
XRHSTOS 35 2610345698
DHMHTRHS 25 2610345695
GIORGOS 45 2610345692
```

Να γραφεί πρόγραμμα το οποίο να ρωτά το χρήστη για το όνομα, την ηλικία και το τηλέφωνο ενός φίλου του και να το γράφει στο αρχείο **my\_friends.txt**.

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

int main()
{
    FILE *writeFile;

    char phone[20];
    char name[25];
    int age;

    writeFile = fopen ("my_friends.txt", "a");
    printf("Please enter the name:");
    scanf("%s",name);
    printf("Please enter the age:");
    scanf("%d",&age);
    printf("Please enter the phone number:");
    scanf("%s",phone);

    fprintf(writeFile, "%s %d %s \n",name, age, phone);
    fclose(writeFile);
    return 0;
}
```

```
Please enter the name:DIONISIS
Please enter the age:24
Please enter the phone number:261087923
```

```
my_friends.txt
XRHSTOS 35 2610345698
DHMHTRHS 25 2610345695
GIORGOS 45 2610345692 32766
DIONISIS 24 261087923
```

# Παράδειγμα

Γράψε ένα πρόγραμμα το οποίο να ρωτά το χρήστη για το όνομα κάποιου φίλου του και να του τυπώνει τον αριθμό του τηλεφώνου του ο οποίος είναι αποθηκευμένος στο αρχείο `my_friends.txt`, στην οθόνη.

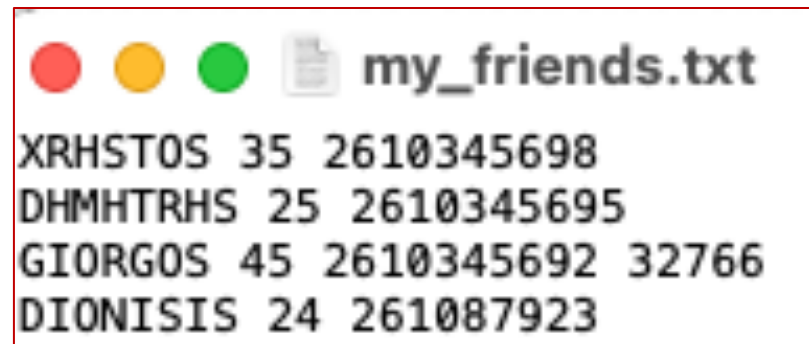
Αν δεν υπάρχει το όνομα να του βγάζει μήνυμα λάθους ότι το όνομα αυτό δεν είναι καταχωρημένο.



# Παράδειγμα

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void get_name_to_find(char *name);
void get_phone_number(char *name);
int main()
{
    char name_to_find[25];
    get_name_to_find(name_to_find);
    get_phone_number(name_to_find);

    return EXIT_SUCCESS;
}
```



The screenshot shows a terminal window with a title bar containing three colored circles (red, yellow, green) and a document icon followed by the filename 'my\_friends.txt'. The terminal output displays the contents of the file, which is a list of names, ages, and phone numbers.

```
XRHSTOS 35 2610345698
DHMHTRHS 25 2610345695
GIORGOS 45 2610345692 32766
DIONISIS 24 261087923
```

```
Search for the name:XRHSTOS
Name: XRHSTOS, Age:35, Phone: 2610345698
```

# Παράδειγμα

```
void get_phone_number(char *name_to_find){
    FILE *readfile;
    int age;
    char phone[20];
    char name[25];
    readfile = fopen ("my_friends.txt", "r");

    if(!readfile){
        printf("Error opening the file");
        return ;
    }

    while( fscanf(readfile,"%s %d %s",name,&age,phone)!=EOF ){
        if(strcmp(name_to_find,name)==0)
            printf("Name: %s, Age:%d, Phone: %s \n", name,age,phone);
    }
    fclose(readfile);
    return;
}

void get_name_to_find(char *name){
    printf("Search for the name:");
    scanf("%s",name);
}
```

# Ευχαριστώ για την προσοχή σας

## ■ Επικοινωνία

- Skype: **fidas.christos**
- Email: **[fidas@upatras.gr](mailto:fidas@upatras.gr)**
- Phone: **2610 – 996491**
- Web: **<http://cfidas.info>**

- **Ώρες γραφείου:** Τετάρτη & Παρασκευή 11:00-13:00

### Join Zoom Meeting

**<https://upatras-gr.zoom.us/j/95080297961?pwd=MzRta0JRd3ZwVEVrREZNc09qbG1Zdz09>**

## Άμεση Επικοινωνία μέσω Skype



**SkypeID:**  
**fidas.christos**

Το υλικό της διάλεξης είναι διαθέσιμο στο eclass

- **<https://eclass.upatras.gr/>**

# ΔΙΑΔΙΚΑΣΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

7<sup>η</sup> Εβδομάδα: Διαχείριση Αρχείων

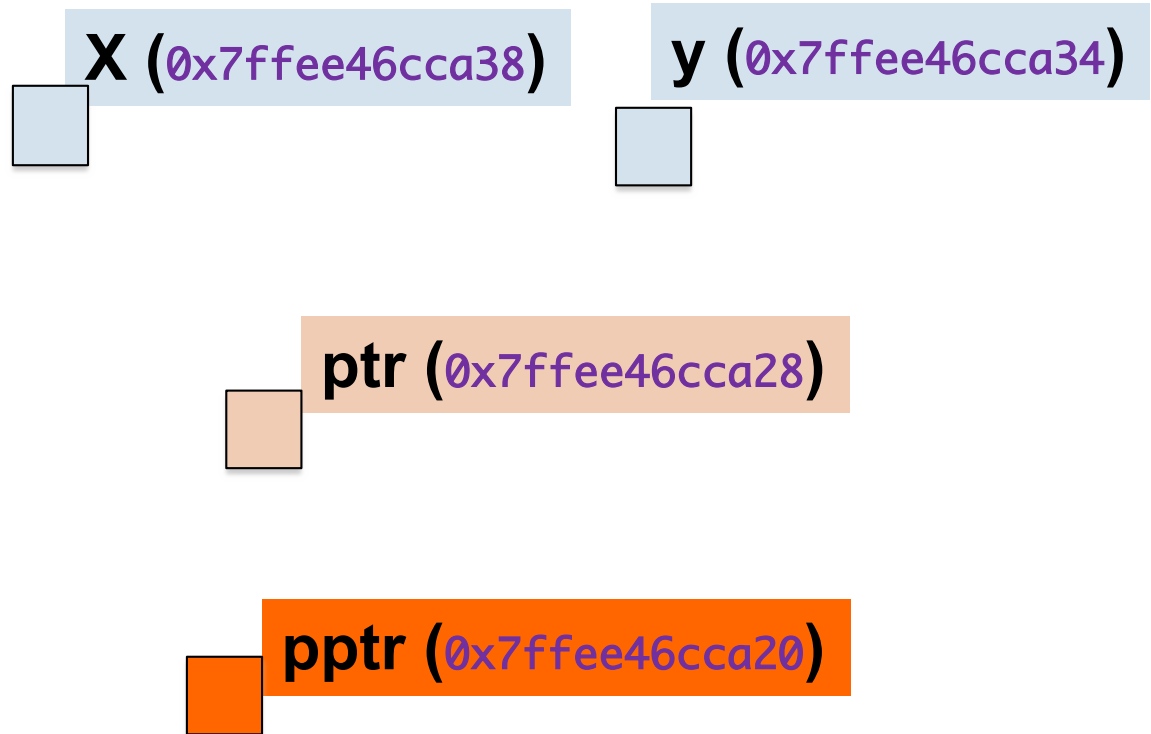
# Αναφορές

Οι διαφάνειες της διάλεξης στηρίζονται, εν μέρει, σε υλικό παραδόσεων παλαιότερων ετών του **Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογία Υπολογιστών του Πανεπιστημίου Πατρών** καθώς και του **Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου**.

# Δείκτη σε δείκτη (επανάληψη)

37

```
int main()
{
    int x;
    int y;
    int *ptr;
    int **pptr;
    return 0;
}
```



# Δείκτης σε δείκτη (επανάληψη)

38

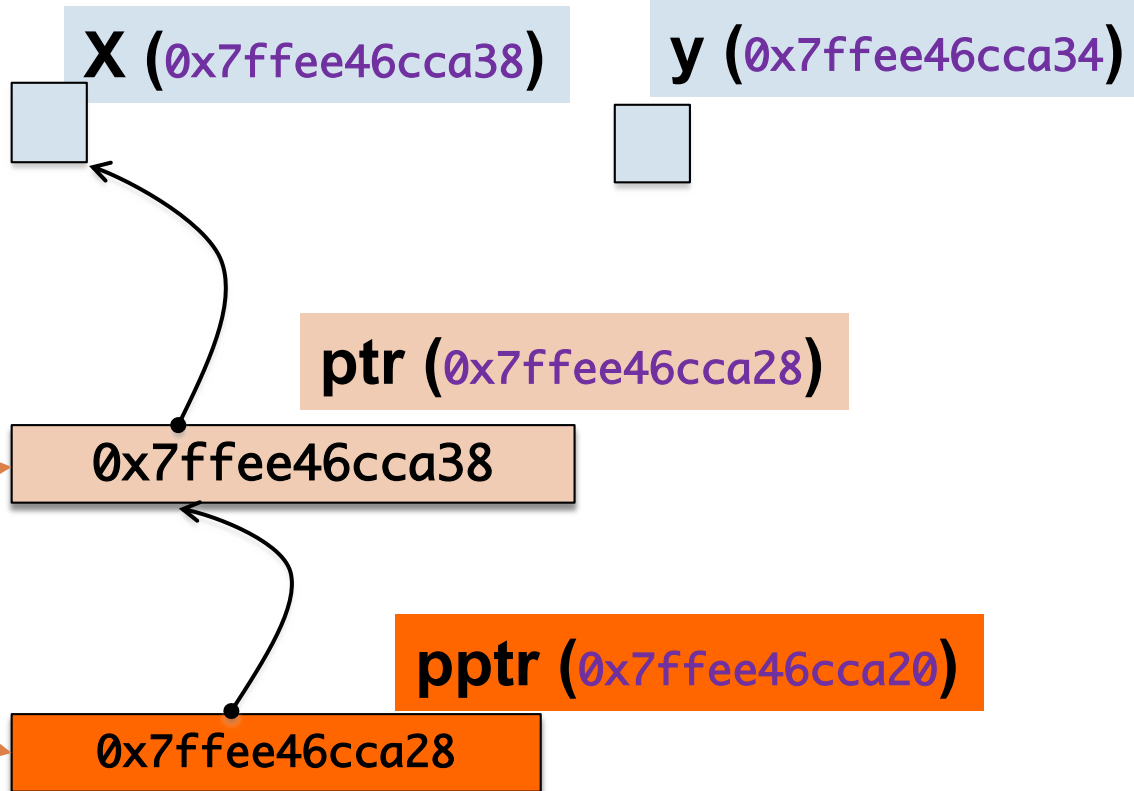
```
int main()
{
    int x;
    int y;
    int *ptr;
    int **pptr;
```

```
    ptr=&x;
```

```
    pptr=&ptr;
```

```
    return 0;
```

```
}
```



# Δείκτης σε δείκτη (επανάληψη)

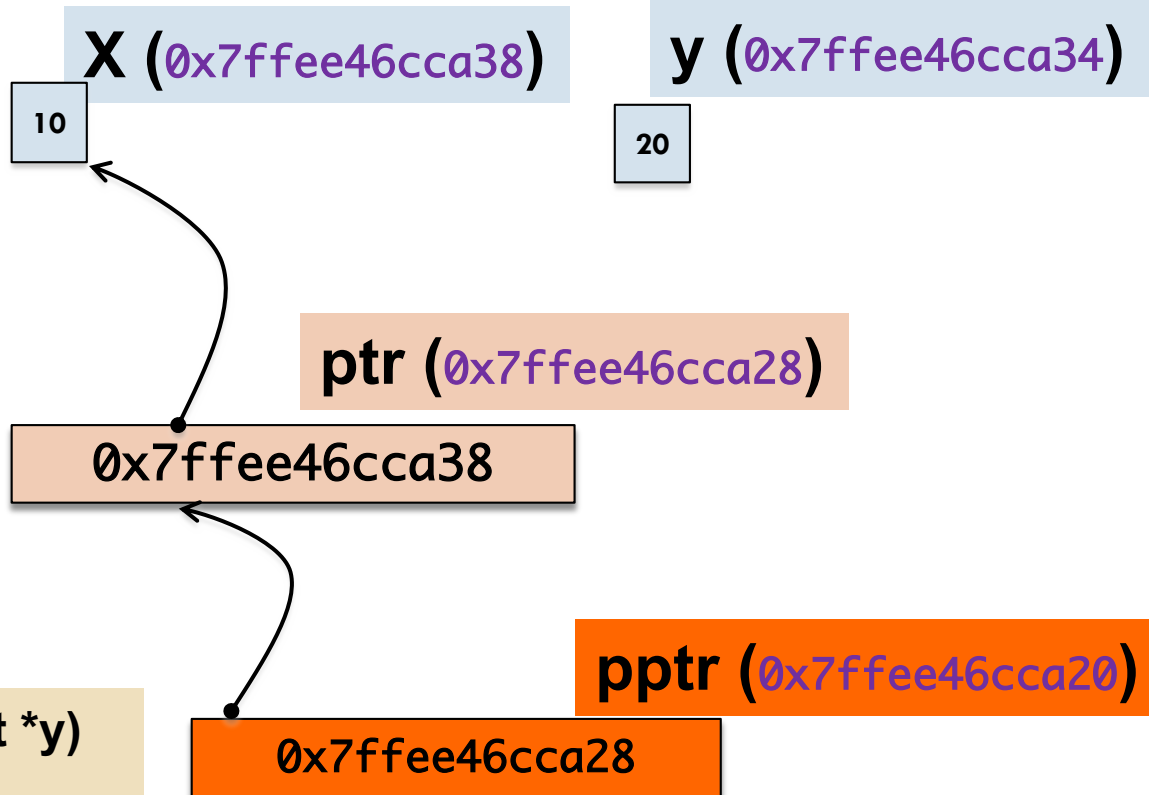
39

Γράψτε μια συνάρτηση η οποία θα αλλάξει τον δείκτη ptr για να δείχνει στην μεταβλητή y

```
void change_pointer (int *px, int *y);
```

```
int main()  
{  
    int x=10;;  
    int y=20;  
    int *ptr;  
    int **pptr;  
  
    ptr=&x;  
    pptr=&ptr;
```

```
change_pointer(ptr,&y);  
return 0;  
}
```



```
void change_pointer (int *px, int *y)
```

```
{  
    px=y;  
    return;
```

Σωστό ή λάθος;

```
}
```



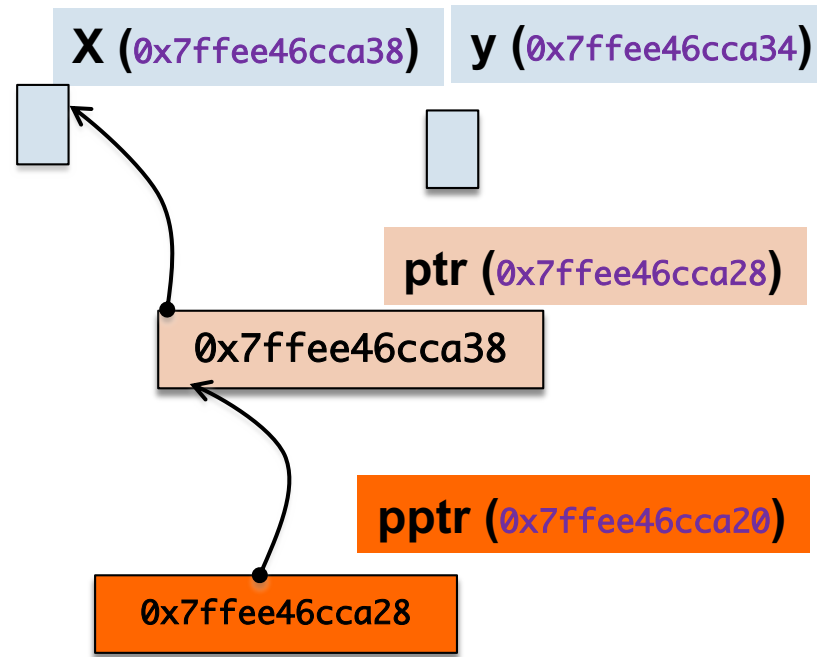
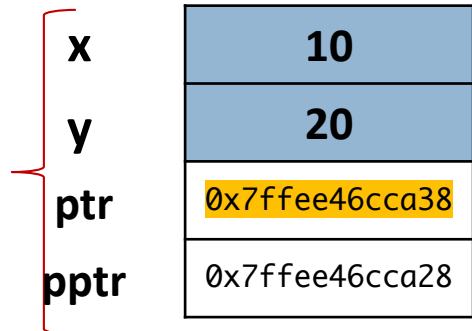
# Δείκτης σε δείκτη

40

```
int main()
{
    int x=10;
    int y=20;
    int *ptr;
    int **pptr;

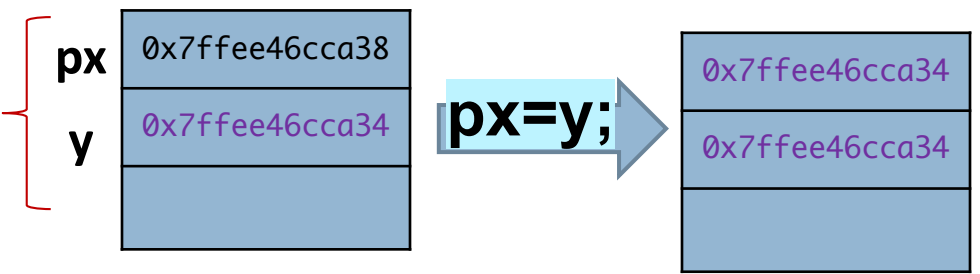
    ptr=&x;
    pptr=&ptr;

    change_pointer(ptr,&y);
    return 0;
}
```



```
void change_pointer (int *px, int *y)
{
    px=y;
    return;
}
```

Η αλλαγή γίνεται στο αντίγραφο του δείκτη μέσα στη συνάρτηση (call by value). Ο ptr στη main θα συνεχίσει να δείχνει την x



# Δείκτης σε δείκτη

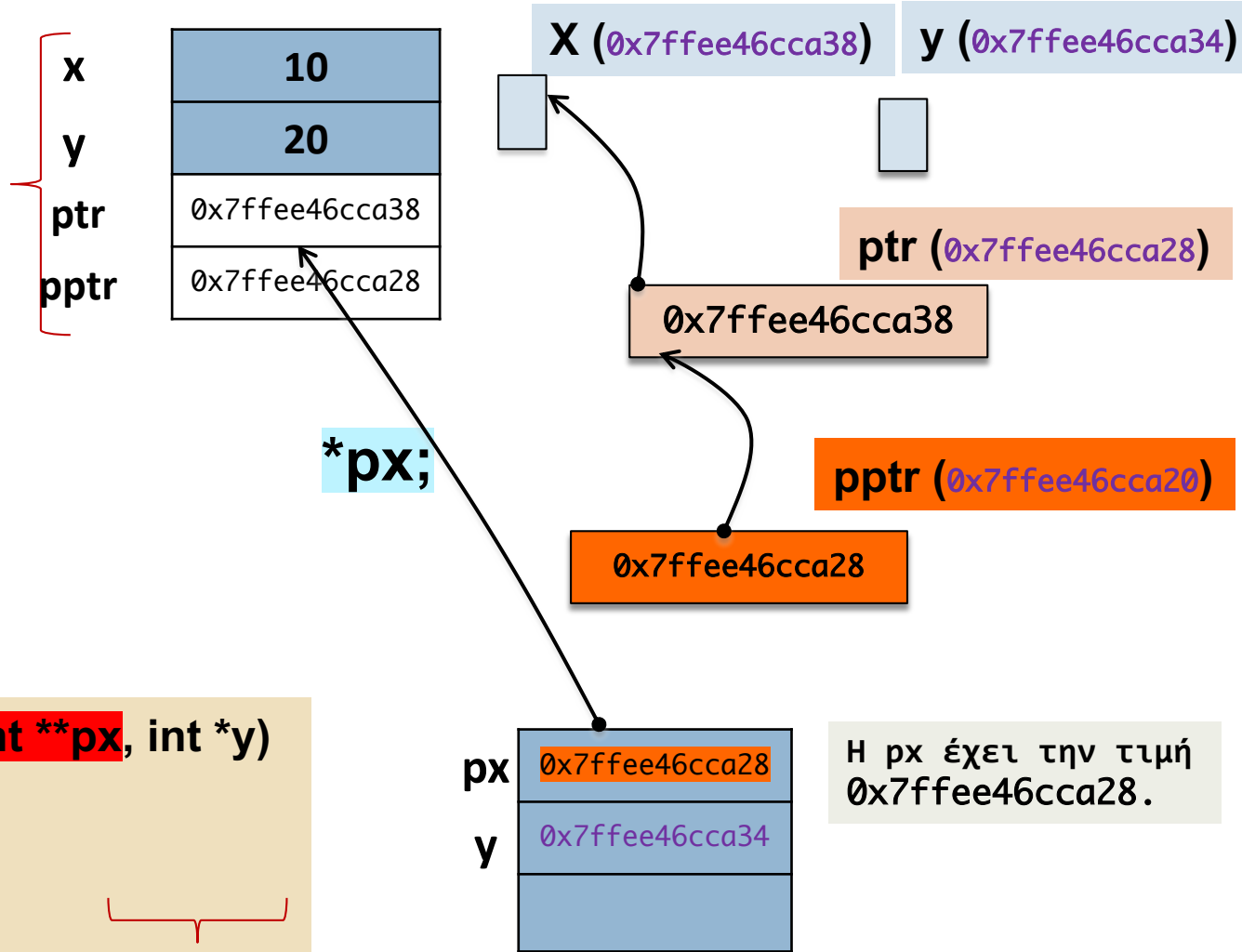
41

```
int main()
{
    int x=10;
    int y=20;
    int *ptr;
    int **pptr;

    ptr=&x;
    pptr=&ptr;

    change_pointer(pptr,&y);
    return 0;
}
```

```
void change_pointer (int **px, int *y)
{
    *px=y;
    return;
}
```



# Δείκτης σε δείκτη

42

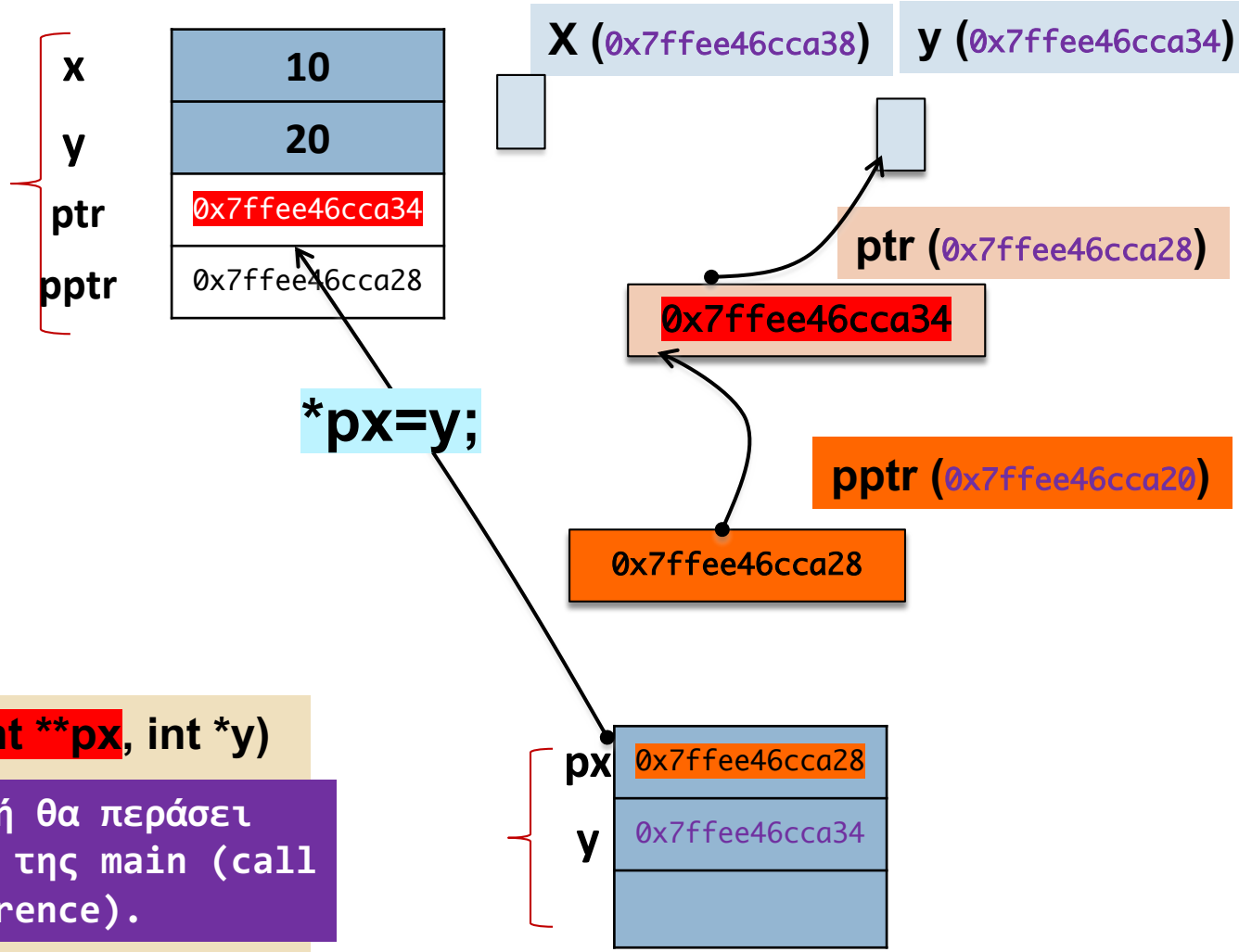
```
int main()
{
    int x=10;
    int y=20;
    int *ptr;
    int **pptr;

    ptr=&x;
    pptr=&ptr;

    change_pointer(pptr,&y);
    return 0;
}
```

```
void change_pointer (int **px, int *y)
{
    *px=y;
    return;
}
```

Η αλλαγή θα περάσει στη ptr της main (call by reference).



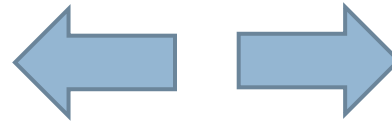
# Δείκτης σε δείκτη

43

```
int main()
{
    int x;
    int y;
    int *ptr;
    int **pptr;

    ptr=&x;
    pptr=&ptr;
```

```
change_pointer(pptr,&y);
return 0;
}
```



Και στις δυο περιπτώσεις τη τιμή του px θα είναι η διεύθυνση του ptr

```
int main()
{
    int x;
    int y;
    int *ptr;

    ptr=&x;
```

```
change_pointer(&ptr,&y);
return 0;
}
```

```
void change_pointer (int **px, int *y)
{
    *px=y;
    return;
}
```

```

#include <stdio.h>
#include <stdlib.h>
void dosomething(int );
int main(void) {
    int a ;

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

    return EXIT_SUCCESS;
}

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

```

```

#include <stdio.h>
#include <stdlib.h>
void dosomething(int *);
int main(void) {
    int a ;

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

    return EXIT_SUCCESS;
}

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

```

**Κώδικας**

**A**

Βρείτε τις διαφορές

**Κώδικας**

**B**

# Κώδικας A

```
C:\Users\paliu\OneDrive - University of Patras\course
a function: 5
a updated function: 6
a main: 5

-----
Process exited after 0.01078 seconds with return code 0
Press any key to continue . . .
```

```
#include <stdio.h>
#include <stdlib.h>
void dosomething(int );
int main(void) {
    int a ;

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

    return EXIT_SUCCESS;
}

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

# Κώδικας B

```
C:\Users\paliu\OneDrive - University of Patras\course
a function: 5
a updated function: 6
a main: 6

-----
Process exited after 0.02233 seconds with return code 0
Press any key to continue . . .
```

```
#include <stdio.h>
#include <stdlib.h>
void dosomething(int *);
int main(void) {
    int a ;

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

    return EXIT_SUCCESS;
}

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

# Σαφέστερος κώδικας με postfix\_ptr στα ονόματα

```
#include <stdio.h>
#include <stdlib.h>
void dosomething(int *);
int main(void) {
    int a ;

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

    return EXIT_SUCCESS;
}

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

# By reference

FILE \* → FILE \* \*

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

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

int main(void) {
    FILE *p;

    fopen(&p);

    filecomplete(p);

    fclose(p);

    return EXIT_SUCCESS;
}
```

```
#include <stdio.h>

void fopen(FILE **p_ptr) {
    *p_ptr = fopen("test.txt", "wt");

    return ;
}
```

Η συνάρτηση `fopen()`  
τροποποιεί το `p`



# Συναρτήσεις που δεν τροποποιούν το p

```
#include <stdio.h>
void filecomplete(FILE *p) {
    int i;
    for (i = 0 ; i < 10 ; i++) {
        fprintf(p, "hello %d\n", i);
    }
    return ;
}
```

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

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

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

```
int main(void) {
    FILE *p;

    p = fopen();

    filecomplete(p);

    fclose(p);
    return EXIT_SUCCESS;
}
```

```
FILE * fopen(void) {
    FILE *p;

    p =fopen("test.txt", "wt");

    return p;
}
```

`char *fgets(char *str, int n, FILE *stream)`

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define LENGTH 80

void clearword(char * word) ;

int main(void) {

    FILE * textin ;
    char word[LENGTH];

    if ( (textin = fopen("mytext2.txt","rt"))==NULL) {
        printf("cannot open file.\n");
        exit(EXIT_FAILURE);
    }

    while (fgets(word, LENGTH, textin)!= NULL) {
        printf("%s\n", word);
        clearword(word);
    }

    fclose(textin);

    return EXIT_SUCCESS;
}
```

## Απόσταση από σημείο αναφοράς

## Ορισμός σημείου αναφοράς

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

51

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

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

**Επιστρέφει τη θέση στην οποία βρισκόμαστε στο τρέχον stream**

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

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

```
int main(void) {
    createfile();

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

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

    return EXIT_SUCCESS;
}
```

Πήγαινε 12 θέσεις από  
την  
Αρχή του αρχείου

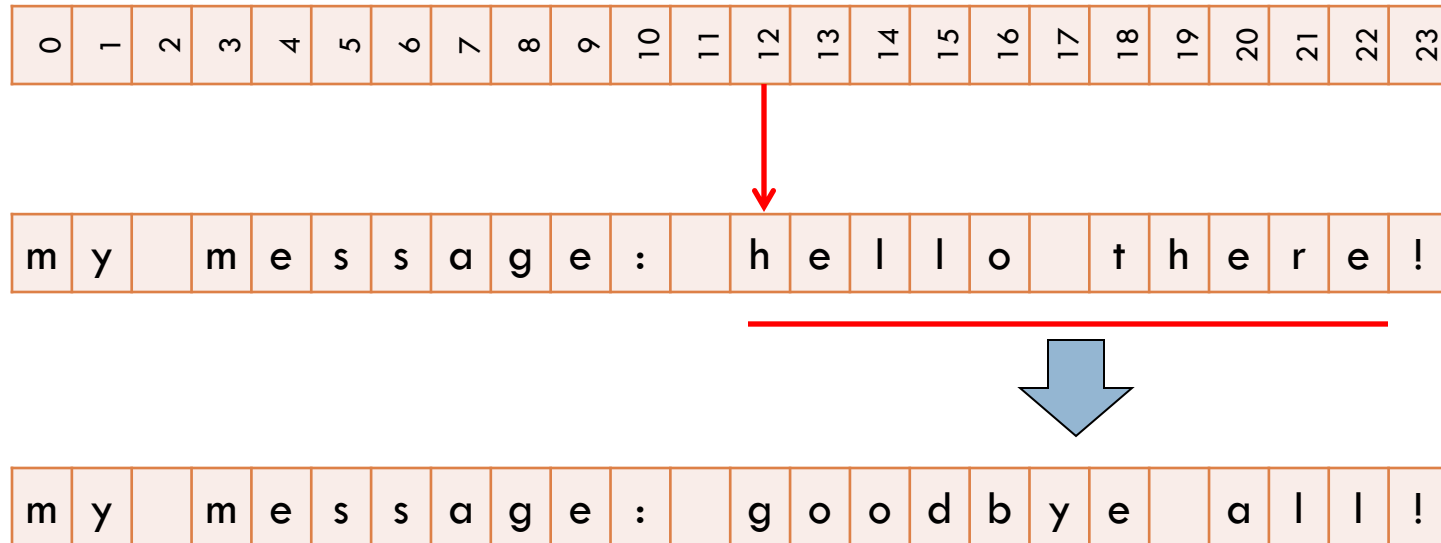
## Δημιουργία αρχείου για εγγραφή

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

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

Άνοιγμα υπάρχοντος  
αρχείου για  
ανάγνωση/ενημέρωση

# Συνέχεια από την προηγούμενη διαφάνεια



# long int ftell(FILE \*)

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

void createfile (void);
void replaceinfile (void);
long int getfilesize(void);

int main(void) {
    long int length;
    createfile( );
    printf("file created\n");
    system("pause");
    replaceinfile( );
    printf("file updated\n");
    length = getfilesize( );
    printf("Contains: %ld chars\n",
           length);

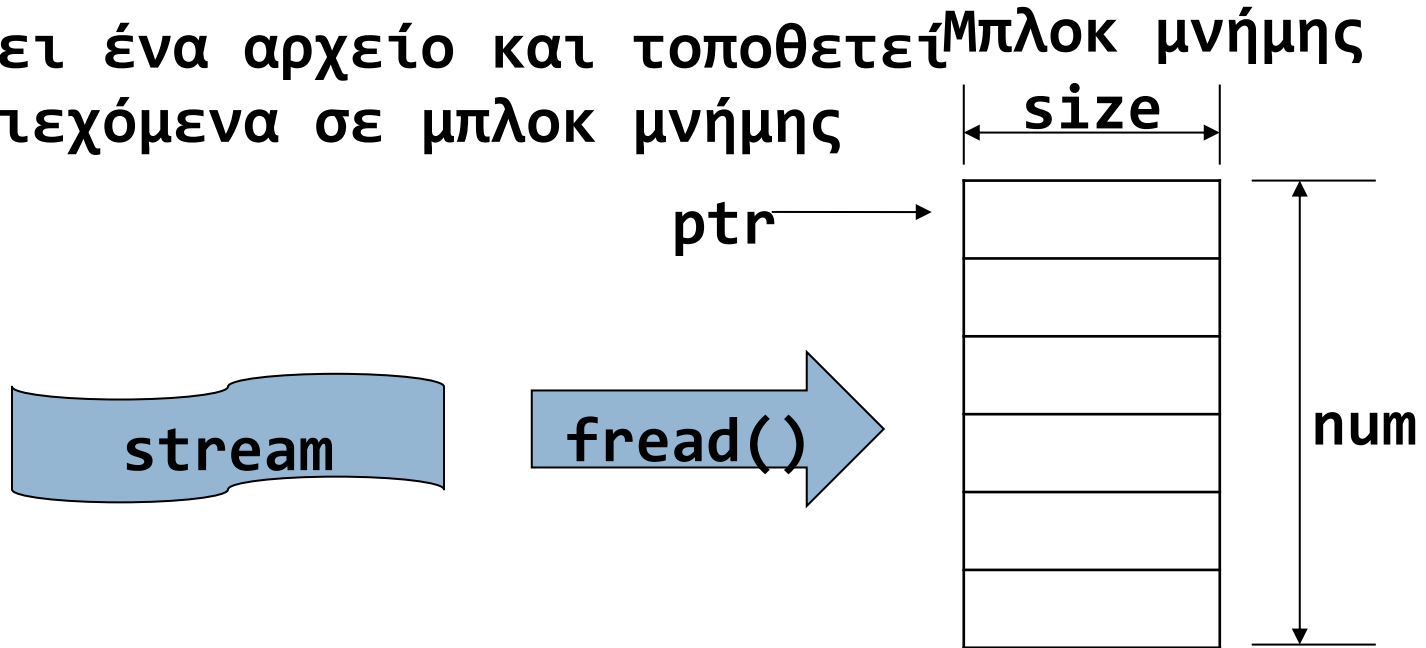
    return EXIT_SUCCESS;
}
```

```
void replaceinfile(void) {
    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);
    return ;
}
```

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

```
size_t fread ( void * ptr, size_t size, size_t num, FILE * stream );
```

Διαβάζει ένα αρχείο και τοποθετεί τα περιεχόμενα σε μπλοκ μνήμης



**ptr:** διεύθυνση πρώτης θέσης του μπλοκ μνήμης

**size:** μέγεθος κάθε στοιχείου (σε bytes)

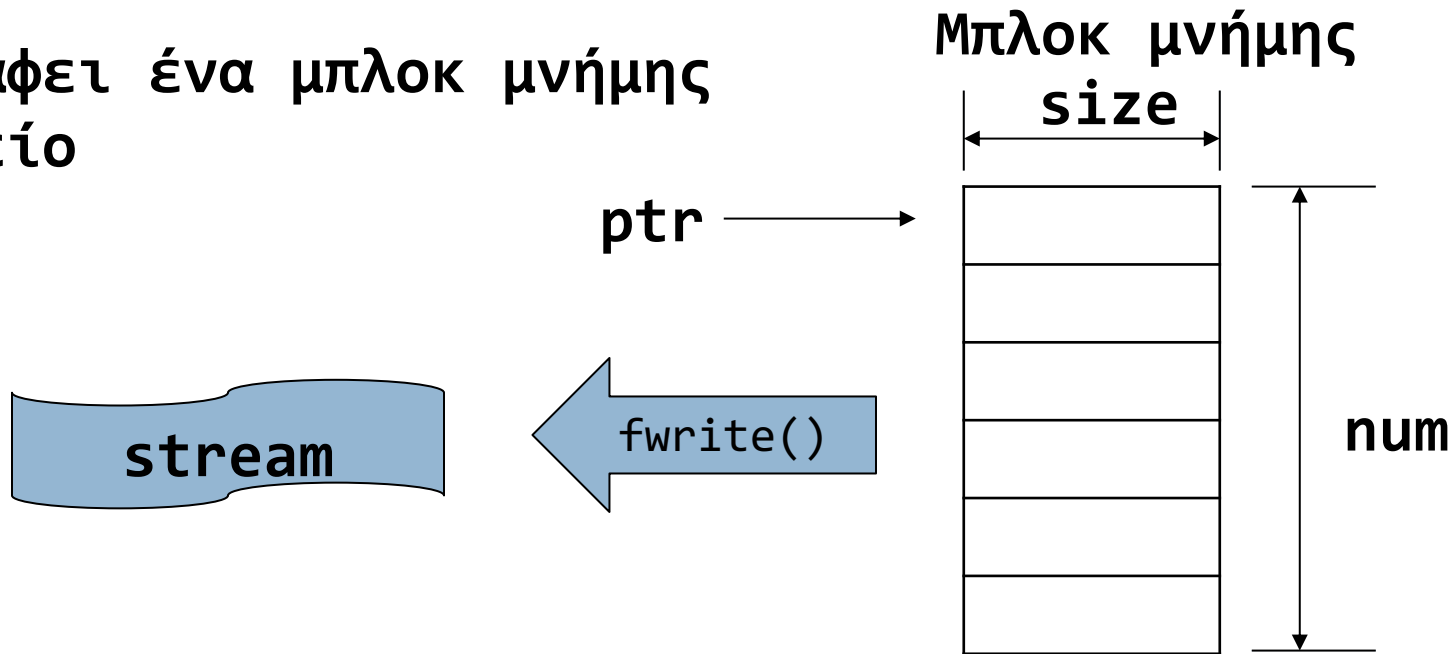
**num:** πλήθος στοιχείων

Επιστρέφει το πλήθος των στοιχείων που αναγνώστηκαν.  
(αν διάφορο του num, έχει συμβεί λάθος ή EOF<sup>5</sup>)



```
size_t fwrite ( const void * ptr, size_t size,  
size_t num, FILE * stream );
```

Αντιγράφει ένα μπλοκ μνήμης  
σε αρχείο



**ptr:** διεύθυνση πρώτης θέσης του μπλοκ μνήμης

**size:** μέγεθος κάθε στοιχείου (σε bytes)

**num:** πλήθος στοιχείων

Επιστρέφει το πλήθος των στοιχείων που εγγράφηκαν.  
(αν διάφορο του num, έχει συμβεί λάθος)

```
void * memset ( void * ptr, int value, size_t num );
```

```
#include <stdio.h>
#include <string.h>
#define SIZE 20
int main(void) {
    int i;
    char bytes[SIZE];

    memset(bytes, 127, SIZE);

    for (i=0; i<SIZE; i++) {
        printf("%hhx ", bytes[i]);
    }

    bytes[SIZE-1] = '\\0';
    memset(bytes, 'Q', SIZE-1);
    printf("\\n%s", bytes);

    return 0;
}
```

```
C:\Users\paliu\OneDrive - University of Patras\courses\PP\1920\remote13\20ex11\Proj
7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f 7f
00000000000000000000000000000000
-----
Process exited after 0.01248 seconds with return value 0
Press any key to continue
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void printwords(char [][][10], int );

int main(void) {
```

```
FILE * mytext ;
char mywords[][10] = {"this", "is", "a", "test."};

printwords(mywords, 4);

if ((mytext=fopen("mydata.bin","wb"))==NULL) {
    printf("Cannot create file\n");
    exit(EXIT_FAILURE);
}
fwrite(mywords, sizeof mywords[0], sizeof mywords/sizeof mywords[0], mytext);
fclose(mytext);

memset(mywords, 0, sizeof mywords);
printwords(mywords, 4);

if ((mytext=fopen("mydata.bin","rb"))==NULL) {
    printf("Cannot read from file\n");
    exit(EXIT_FAILURE);
}
fread(mywords, sizeof mywords[0], sizeof mywords/ sizeof mywords[0], mytext);
fclose(mytext);

printwords(mywords, 4);

return EXIT_SUCCESS;
}
```

```
void printwords(char wd[][10] , int len) {
    int i;

    printf("\nmywords: ");
    for (i=0; i<len; i++) printf("%s ",wd[i]);

    return;
}
```

```
C:\Users\paliu\OneDrive - University of Patras\courses\PP\1920\remote13\20ex06
mywords: this is a test.
mywords:
mywords: this is a test.
-----
Process exited after 0.01709 seconds with return value 0
Press any key to continue . . .
```

# Ευχαριστώ για την προσοχή σας

## ■ Επικοινωνία

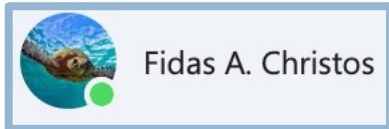
- Skype: **fidas.christos**
- Email: **[fidas@upatras.gr](mailto:fidas@upatras.gr)**
- Phone: **2610 – 996491**
- Web: **<http://cfidas.info>**

- **Ώρες γραφείου:** Τετάρτη & Παρασκευή 11:00-13:00

### Join Zoom Meeting

**<https://upatras-gr.zoom.us/j/95080297961?pwd=MzRta0JRd3ZwVEVrREZNc09qbG1Zdz09>**

## Άμεση Επικοινωνία μέσω Skype



**SkypeID:**  
**fidas.christos**

Το υλικό της διάλεξης είναι διαθέσιμο στο eclass

- **<https://eclass.upatras.gr/>**