

# ΕΠΛ 033: ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΓΙΑ ΜΗΧΑΝΙΚΟΥΣ

Μάριος Belk, Τμήμα Πληροφορικής, Πανεπιστήμιο Κύπρου

Email: [belk@cs.ucy.ac.cy](mailto:belk@cs.ucy.ac.cy)





# Επικοινωνία με Πρόγραμμα Χρήση Αρχείων

# Επικοινωνία με Πρόγραμμα

## □ Επικοινωνία με ένα πρόγραμμα

- ▣ Εισαγωγή δεδομένων
- ▣ Εξαγωγή δεδομένων

γίνεται ουσιαστικά με δύο τρόπους

## □ Καθώς τρέχει το πρόγραμμα

- ▣ Ο χρήσης δίνει δεδομένα από το πληκτρολόγιο
- ▣ Ο χρήστης πληροφορείται πότε και τι να δώσει από την οθόνη

## □ Πληροφορίες από/σε αρχείο

- ▣ Με χρήση ανακατεύθυνσης (redirection)
- ▣ Με χρήση εντολών `fprintf`, `fscanf`

# Είσοδος / Έξοδος με Ανακατεύθυνση

- Ανακατεύθυνση εισόδου
  - ▣ `myprogram < in`
  - ▣ `in`: αρχείο τύπου ASCII που περιέχει πληροφορίες εισόδου (American Standard Code for Information Interchange)
- Ανακατεύθυνση εξόδου
  - ▣ `myprogram > out`
  - ▣ `out`: αρχείο τύπου ASCII που αποθηκεύονται τα αποτελέσματα εξόδου
- Ανακατεύθυνση εισόδου και εξόδου
  - ▣ `myprogram <in >out`

# Παράδειγμα

```
/* programma revnum.c */
#include<stdio.h>
int main(){

float lires, rate;

scanf("%f %f", &lires, &rate);

printf("Oi %f lires einai %f Euro\n", lires, lires*rate);

return 0;
}
```

*Τι πρέπει να περιέχει το **in** πριν την εκτέλεση  
και τι περιέχει το **out** μετά την εκτέλεση  
`revnum < in > out` ?*

# Δεδομένα από Συγκεκριμένο Αρχείο

- Ένα πρόγραμμα μπορεί να καθορίζει το αρχείο από το οποίο θα διαβάσει ή θα αποθηκεύσει δεδομένα
- Πρώτα ορίζεται ένα *δείκτης τύπου αρχείου* ο οποίος θα δείχνει στο συγκεκριμένο αρχείο, π.χ.:
  - ▣ FILE \*inp (δείκτης στο αρχείο εισόδου)
  - ▣ FILE \*outp (δείκτης στο αρχείο εξόδου)
- Τα αρχεία “ανοίγονται” στο πρόγραμμα με την εντολή  
fopen(“file\_name”, “access\_type”)
  - ▣ inp = fopen(“input.dat”, “r”)
  - ▣ outp = fopen(“output.dat”, “w”)
  - ▣ (Υπάρχει και το access\_type “a”)

# Επιλογές χρήσης (“ανοίγματος”) αρχείου

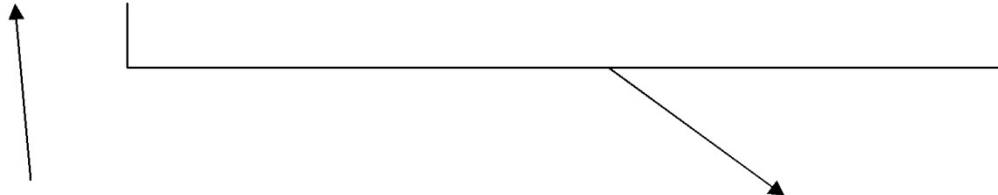
6

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.

# Εντολή fprintf

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

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



Δείκτης σε αρχείο εξόδου

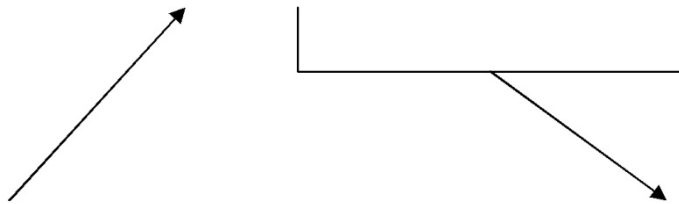
Όπως printf



# Εντολή fscanf

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

π.χ `fscanf(inp, "%d", &num )`



Δείκτης σε αρχείο εισόδου Όπως `scanf`

# Κλείσιμο Αρχείων

- Όταν δεν χρειαζόμαστε τα αρχεία, τότε τα “κλείνουμε” με την εντολή:  
`fclose(δείκτης στο αρχείο)`
  - ▣ `fclose(inp);`
  - ▣ `fclose(outp);`
- Αν δεν κλείσουμε τα αρχεία πριν τον τερματισμό του προγράμματος, τότε τα αρχεία κλείνουν ταυτόχρονα με την εκτέλεση της `return 0` της `main()`

# Παράδειγμα

```
/* programma revnum.c */
#include<stdio.h>
int main(){
float lires, rate;
FILE *inp, *outp;

inp = fopen("input.dat", "r");
outp = fopen("output.dat", "w");

fscanf(inp, "%f %f", &lires, &rate);
fprintf(outp, "Oi %f Lires einai %f Euro\n", lires, lires*rate);

fclose(inp);
fclose(outp);

return 0;
}
```

# Παράδειγμα

11

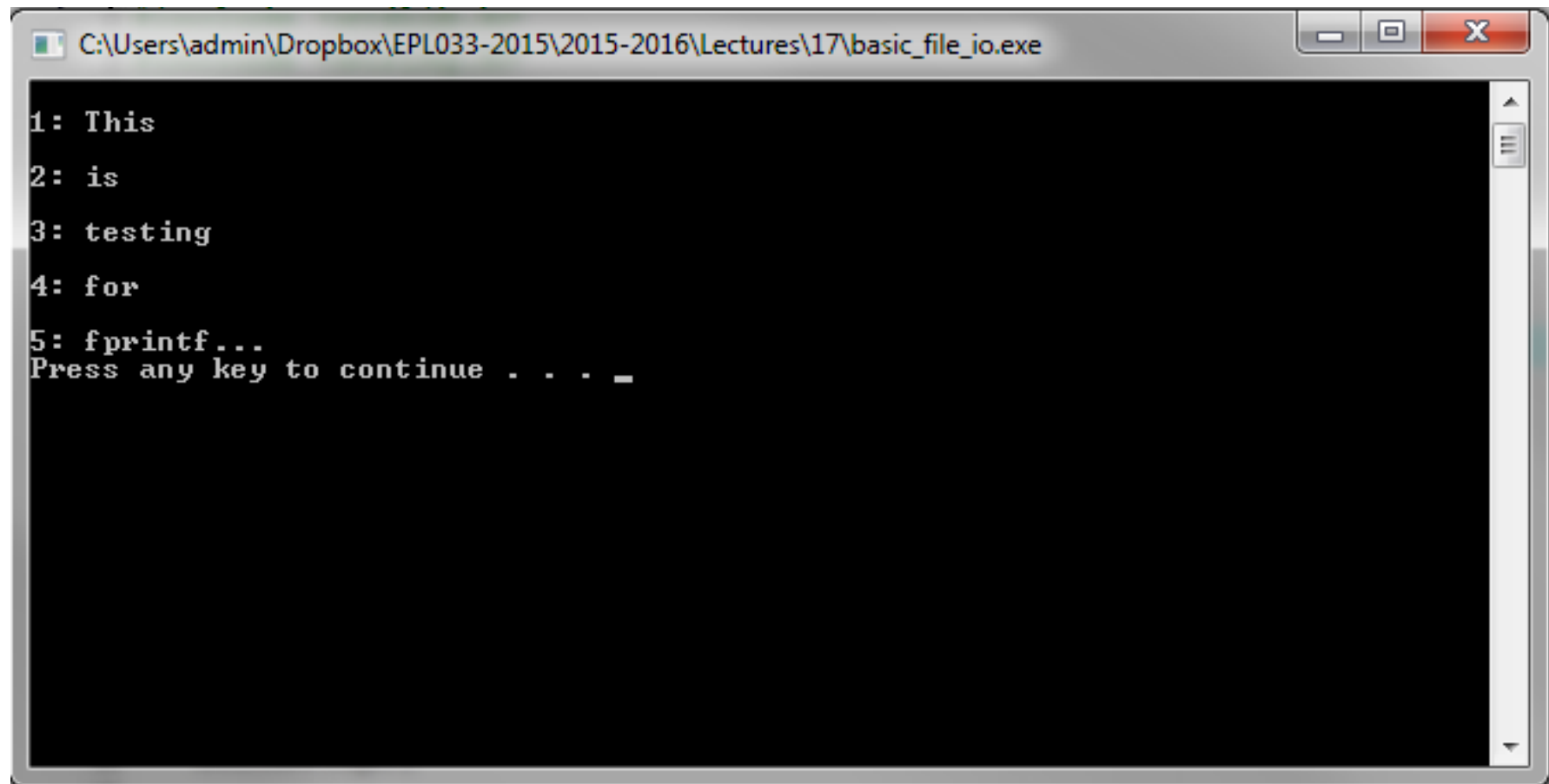
```
FILE *fp;
char buff[255];
int i;

fp = fopen("test.txt", "w+");

if (!fp) {
    printf("Error opening the file!\n");
    exit(-1);
}

fprintf(fp, "This is testing for fprintf...\n");
fclose(fp);
```

```
fp = fopen("test.txt", "r");
if (!fp) {
    printf("Error opening the file!\n");
    exit(-1);
}
i=0;
while(!feof(fp))
{
    if (fscanf(fp, "%s", buff)!=EOF)
        printf("\n%d: %s\n", i+1, buff );
    i++;
}
fclose(fp);
```



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\admin\Dropbox\EPL033-2015\2015-2016\Lectures\17\basic\_file\_io.exe. The window contains the following text:

```
1: This
2: is
3: testing
4: for
5: fprintf...
Press any key to continue . . . _
```

```
student_detail stu_data;
FILE *fp;
// read students detail
printf("Student id:");
scanf("%d",&stu_data.id);
printf("Student name:");
scanf("%s",stu_data.name);
printf("Percentage:");
scanf("%f",&stu_data.percentage);
printf("\nCollege id:");
scanf("%d",&stu_data.clg_data.college_id);
printf("College name:");
scanf("%s",stu_data.clg_data.college_name);
```

```
fp = fopen("test.txt", "w+");

fprintf(fp, "%d\n", stu_data.id);
fprintf(fp, stu_data.name);
fprintf(fp, "\n");
fprintf(fp, "%.2f\n", stu_data.percentage);
fprintf(fp, "%d\n", stu_data.clg_data.college_id);
fprintf(fp, stu_data.clg_data.college_name);
fprintf(fp, "\n");

fclose(fp);
```



```
fp = fopen("test.txt", "r");

fscanf(fp, "%d", &stu_data.id);
fscanf(fp, "%s", stu_data.name);
fscanf(fp, "%f", &stu_data.percentage);

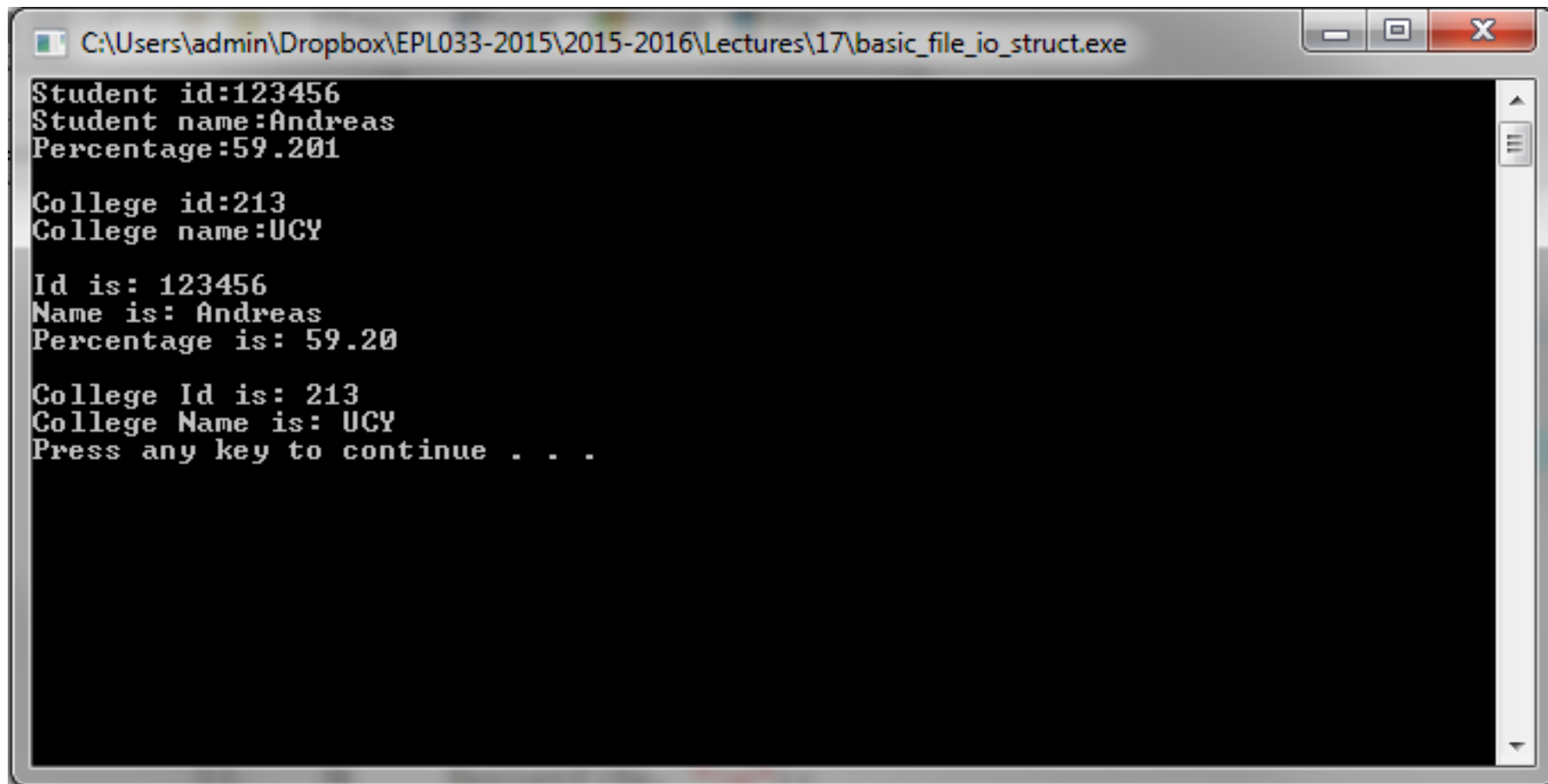
fscanf(fp, "%d", &stu_data.clg_data.college_id);
fscanf(fp, "%s", stu_data.clg_data.college_name);

fclose(fp);
```

Test.txt

```
123456
Andreas
59.20
213
UCY
```

...print struct elements...

A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\admin\Dropbox\EPL033-2015\2015-2016\Lectures\17\basic\_file\_io\_struct.exe. The window contains the following text:

```
Student id:123456  
Student name:Andreas  
Percentage:59.201  
  
College id:213  
College name:UCY  
  
Id is: 123456  
Name is: Andreas  
Percentage is: 59.20  
  
College Id is: 213  
College Name is: UCY  
Press any key to continue . . .
```