

Προγραμματισμός

Αρχεία



Η Βιβλιοθήκη Εισόδου/Εξόδου

- Στο `stdio.h`
 - Δηλώνεται ο τύπος δεδομένων `FILE` (με `typedef`) με όλες τις πληροφορίες που χρειάζονται για προσπέλαση του αρχείου από τις αντίστοιχες συναρτήσεις
 - Ορίζονται 3 αρχεία
 - `stdin`, `stdout`, `stderr` (θα δούμε ποια είναι αυτά σε λίγο)
 - Ορίζονται οι τιμές των σταθερών
 - `NULL`
 - `EOF` (End Of File)
 - Δηλώνονται συναρτήσεις προσπέλασης αρχείων



Αρχεία

- Η C βλέπει κάθε αρχείο σαν μια σειρά από bytes
 - Κάθε αρχείο «κλείνει» με end-of-file marker
- Μπορούμε να ανοίξουμε ένα αρχείο για διάβασμα, είτε για γράψιμο
- Το άνοιγμα του αρχείου συνεπάγεται την επιστροφή ενός δείκτη σε FILE structure



Άνοιγμα Αρχείων

- **FILE *fp;**
 - Δημιουργεί έναν **FILE** pointer με όνομα **fp**
- **FILE *fopen(const char **fname*, const char **mode*);**
 - Δύο ορίσματα:
 - filename: το όνομα του αρχείου που θα ανοίξει
 - mode: Καθορίζει αν το αρχείο ανοιχτεί για διάβασμα ή γράψιμο
 - Η fopen επιστρέφει έναν **FILE** pointer στο αρχείο
 - Σε περίπτωση αποτυχίας επιστρέφει NULL

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



File open modes

mode			description	starts..
r	rb		open for reading	beginning
w	wb		open for writing (creates file if it doesn't exist). Deletes content and overwrites the file.	beginning
a	ab		open for appending (creates file if it doesn't exist)	end
r+	rb+	r+b	open for reading and writing	beginning
w+	wb+	w+b	open for reading and writing. Deletes content and overwrites the file.	beginning
a+	ab+	a+b	open for reading and writing (append if file exists)	end



Ανίχνευση λαθών

```
if ((output_file = fopen("output_file", "w")) == (FILE *) NULL)
```

```
    fprintf(stderr, "Cannot open file.\n");
```



Σειριακή προσπάθεια



Δημιουργία αρχείου κειμένου

```
main( )
{
    FILE *fp;
    int index;

    fp = fopen("mytestfile.txt", "w"); /* open for writing */

    for (index = 1; index <= 10; index++)
        fprintf(fp, "Line number %d\n", index);

    fclose(fp); /* close the file before ending program */
}
```



Ανάγνωση και προβολή αρχείου κειμένου

```
int main( )
{
    FILE *funny;
    int c;
    funny = fopen("file2read.txt","r");
    if (funny == NULL)
    {
        printf("File doesn't exist\n");
        return 1;
    }
    do
    {
        c = getc(funny); /* get a character from the file */
        putchar(c); /* display it on the monitor */
    } while (c != EOF); /* until EOF */
    fclose(funny);
}
```



Αντιγραφή αρχείου σε άλλο

```
void filecopy(FILE *ifp, FILE *ofp)
{
    int c;
    while ((c = getc(ifp)) != EOF)
        putc(c, ofp);
}
```

```
FILE *ifp = fopen("input.txt", "r");
FILE *ofp = fopen("output.txt", "w");
```

```
filecopy(ifp, ofp);
```

```
fclose(ifp);
fclose(ofp);
```



Append a file

```
#include <stdio.h>

int main()
{
    FILE *fp file = fopen("file.txt","a");
    fprintf(fp,"%s","This is just an example :);");
    /*append some text*/
    fclose(fp);
    return 0;
}
```



Παράδειγμα – ανάγνωση λέξεων

υποθέτει πως 1 γραμμή <999 chars

```
void somefunction(char *file)
{
    FILE *fp;
    char s[1000];

    if ((fp = fopen(file, "r")) == NULL)
    {
        printf("Can not open the file %s\n",file);
        return;
    }

    while (fscanf(fp,"%s",s) != EOF)
    {
        printf("%s",s);
    }
    fclose(fp);
}
```



Κλείσιμο Αρχείων

- **Ότι ανοίγει, πρέπει να κλείνει**
 - Το πρόγραμμα δεν πρέπει να αφήνει ανοιχτά αρχεία
 - μπορεί να μην περάσουν όλες οι αλλαγές σας
 - δυσχεραίνει η πρόσβαση σε αυτά από άλλα προγράμματα
- **fclose(*FILE pointer*)**
 - κλείνει το αρχείο
 - Επιστρέφει 0 αν είναι επιτυχής και EOF διαφορετικά
 - Γίνεται αυτόματα με το τέλος του προγράμματος
 - Είναι καλή πρακτική να γίνεται από τον προγραμματιστή



Ανάγνωση/Εγγραφή byte προς byte

- **fgetc** (αντίστοιχη της *getchar*)
 - Διαβάζει έναν χαρακτήρα από αρχείο
 - Όρισμα: **FILE** pointer
 - **fgetc(stdin)** ισοδύναμο με **getchar()**
- **fputc** (αντίστοιχη της *putchar*)
 - Γράφει έναν χαρακτήρα σε αρχείο
 - Όρισμα: **FILE** pointer και ένας χαρακτήρας για να γραφεί
 - **fputc('a', stdout)** ισοδύναμο με **putchar('a')**



Ανάγνωση/Εγγραφή γραμμή προς γραμμή

- **fgets**

- Διαβάζει μια γραμμή από ένα αρχείο
- *char *fgets(char *line, int maxline, FILE *fp)*

- **fputs**

- Γράφει μια γραμμή σε ένα αρχείο
- *int fputs(char *line, FILE *fp)*



Μορφομένη (formatted) ανάγνωση/εγγραφή

- **fscanf / fprintf**
 - Ισοδύναμες των **scanf** και **printf** για αρχεία



Τέλος προσπάθειας

- Οι συναρτήσεις ανάγνωσης επιστρέφουν **EOF**, ή **(FILE *) NULL**, όταν φτάσουν στο τέλος του αρχείου
- Η συνάρτηση:
`int feof ( FILE * stream );`
- επιστρέφει 0 για τέλος αρχείου και διαφορετικά (!0), αλλιώς, Π.χ.

```
while (!feof(ifp))
{
    if (fscanf(ifp, "%s %d", username, &score) != 2)
        break;
    fprintf(ofp, "%s %d", username, score+10);
}
```



EOF pitfall

A common mistake when using `fgetc`, `getc`, or `getchar` is to assign the result to a variable of type `char` *before* comparing it to `EOF`. The following code fragments exhibit this mistake, and then show the correct approach (using type `int`):

Mistake	Correction
<pre>char c; while ((c = getchar()) != EOF) putchar(c);</pre>	<pre>int c; while ((c = getchar()) != EOF) putchar(c);</pre>

- Σύγκριση του `EOF` με έναν από τους 256 χαρακτήρες
- Αλλιώς: check `feof` and `ferror` after `getchar` returns `EOF`.



Standard file pointers in UNIX

FILE *stdin, *stdout, *stderr;

- automatically open to all C programs
- since these files are already open, there is no need to use fopen on them



Τρία Ειδικής Χρήσης Αρχεία

- Το λειτουργικό σύστημα, με το που αρχίζει να τρέχει το πρόγραμμά, **ανοίγει τρία αρχεία** και ορίζει τους δείκτες `stdin`, `stdout`, `stderr` να δείχνουν στις αντίστοιχες δομές `FILE`. Κανονικά :
 - **`stdin`**: αντιστοιχεί στο πληκτρολόγιο! Ότι κουμπί πατιέται θεωρείται ότι διαβάζεται από ένα αρχείο
 - **`stdout`**: αντιστοιχεί στην οθόνη μας. Ότι εγγράφετε σε αυτό το αρχείο εκτυπώνεται στην οθόνη
 - **`stderr`**: αντιστοιχεί επίσης στην οθόνη μας. Χρησιμοποιείτε για μηνύματα λάθους.



Αναδιεύθυνση (redirection) στο Unix

- Μπορούμε να ζητήσουμε από το λειτουργικό να *ανακατευθύνει* τα stdin, stdout, stderr με αρχεία στο δίσκο:
 - a.out < someinputfile
 - Το stdin δεν είναι πλέον το πληκτρολόγιο, αλλά το αρχείο somefile
 - a.out > someoutputfile
 - Το stdout δεν είναι πλέον η οθόνη αλλά το someoutputfile
 - a.out < inputfile > outputfile
 - a.out < inputfile > outputfile >& errfile



Παράδειγμα

```
#include <stdio.h>

int main()
{
    int i;
    FILE *fp;

    if ((fp = fopen("myfirstfile.txt", "w")) == NULL)
    {
        printf("Could not open file.\n");
        return 0;
    }

    fprintf(fp, "Lets write something here.\n");
    for (i = 0; i < 10; ++i)
        fprintf(fp, "%5d, %5d, %5d\n", i, i*i, i*i*i);

    fclose(fp);
    // Check out the contents of the file myfirstfile.txt
}
```

Lets write something here.

0,	0,	0
1,	1,	1
2,	4,	8
3,	9,	27
4,	16,	64
5,	25,	125
6,	36,	216
7,	49,	343
8,	64,	512
9,	81,	729



```

1
2
3 #include <stdio.h>
4
5 int main()
6 {
7     int account = 1;
8     char name[ 30 ];
9     double balance;
10    FILE *cfPtr;    /* cfPtr = clients.dat file pointer */
11
12    if ( ( cfPtr = fopen( "clients.dat", "w" ) ) == NULL )
13        printf( "File could not be opened\n" );
14    else {
15        printf( "Enter the account, name, and balance.\n" );
16        printf( "Enter EOF to end input.\n" );
17        printf( "? " );
18        scanf( "%d%s%lf", &account, name, &balance );
19
20        while (account > 0) {
21            fprintf( cfPtr, "%d %s %.2f\n",
22                    account, name, balance );
23            printf( "? " );
24            scanf( "%d%s%lf", &account, name, &balance );
25        }
26
27        fclose( cfPtr );
28    }
29
30    return 0;
31 }

```

Enter the account, name, and balance.

Enter EOF to end input.

? 100 Jones 24.98

? 200 Doe 345.67

? 300 White 0.00

? 400 Stone -42.16

? 500 Rich 224.62

?

Τυχαία προσπέλαση



Τρέχουσα Θέση

- Μία από τις πληροφορίες που διατηρούνται στις δομές τύπου FILE για κάθε αρχείο είναι η τρέχουσα θέση εγγραφής/ανάγνωσης
- Όταν κάνουμε εγγραφή κειμένου σε ένα αρχείο με τις συναρτήσεις `fputc`, `fputs`, `fprintf` η εγγραφή γίνεται στην τρέχουσα θέση μέσα στο αρχείο.
- Όταν κάνουμε ανάγνωση κειμένου με τις συναρτήσεις `fgetc`, `fgets`, `fscanf` η ανάγνωση γίνεται από την τρέχουσα θέση στο αρχείο.
- Η τρέχουσα θέση αυξάνει ανάλογα με τους χαρακτήρες που έχουμε γράψει ή διαβάσει
- Η τρέχουσα θέση όταν πρωτανοίξουμε ένα αρχείο εξαρτάται από το `mode` στην `fopen`
- Μπορούμε να αλλάξουμε την τρέχουσα θέση με την συνάρτηση `fseek` (αναλυτικά μετά)



Βασικές εντολές

- **fseek()** Sets the position to a desired point in the file
- **ftell()** Gives the current position in the file
- **rewind()** Sets the position to the beginning of the file



Μετακίνηση του δείκτη αρχείου

- **fseek**

- Πάει το δείκτη σε μια συγκεκριμένη θέση
- **fseek(*pointer*, *offset*, *symbolic_constant*);**
 - *pointer* – file pointer
 - *offset* – θέση file pointer (0 η αρχή του αρχείου)
 - *symbolic_constant* – καθορίζει από πού αρχίζουμε να μετράμε
 - **SEEK_SET** – αρχή αρχείου
 - **SEEK_CUR** – τρέχουσα θέση
 - **SEEK_END** – τέλος αρχείου



ΗΥ-150

Προγραμματισμός

Δυναμικά αρχεία



fread() / fwrite()

- `size_t fread(void *ptr, size_t size_of_elements, size_t number_of_elements, FILE *a_file);`
- `size_t fwrite(const void *ptr, size_t size_of_elements, size_t number_of_elements, FILE *a_file);`

Μπορεί να χρησιμοποιηθεί για κάθε μεταβλητή (ακόμη και για απλές ή στατικές με χρήση του &)



Δυναμικά αρχεία

- **fwrite**

- «γράφει» δεδομένα από μια περιοχή μνήμης σε ένα αρχείο

- Παράδειγμα:

fwrite(&number, sizeof(int), 1, myPtr);

- **&number** – περιοχή μνήμης (δείκτης)
- **sizeof(int)** – μέγεθος αντικειμένων
- **1** – αριθμός αντικειμένων (1 για απλές μεταβλητές)
- **myPtr** – αρχείο για μεταφορά

- **fread**

- «διαβάζει» δεδομένα από ένα αρχείο σε μια περιοχή μνήμης

fread(&client, sizeof (int), 1, myPtr);

- Μπορεί να «διαβάσει» ολόκληρο τον πίνακα



Παράδειγμα

```
FILE *fp; fp=fopen("c:\\test.bin", "wb");  
char x[10]="ABCDEFGHIIJ";  
fwrite(x, sizeof(x[0]), sizeof(x)/sizeof(x[0]), fp);  
fclose(fp);
```



Παράδειγμα

```
int main(void)
{
    FILE *fp; size_t count;
    const char *str = "hello\n";
    fp = fopen("sample.txt", "w");
    if(fp == NULL)
    {
        perror("failed to open sample.txt");
        return EXIT_FAILURE;
    }
    count = fwrite(str, 1, strlen(str), fp);
    printf("Wrote %d bytes. fclose(fp) %s.\n", count, fclose(fp) == 0 ?
        "succeeded" : "failed");
    return EXIT_SUCCESS;
}
```



```
1
2
3 #include <stdio.h>
4
5 struct clientData {
6     int acctNum;
7     char lastName[ 15 ];
8     char firstName[ 10 ];
9     double balance;
10 };
11
12 int main()
13 {
14     int i;
15     struct clientData blankClient = { 0, "", "", 0.0 };
16     FILE *cfPtr;
17
18     if ( ( cfPtr = fopen( "credit.dat", "w" ) ) == NULL )
19         printf( "File could not be opened.\n" );
20     else {
21
22         for ( i = 1; i <= 100; i++ )
23             fwrite( &blankClient,
24                 sizeof( struct clientData ), 1, cfPtr );
25
26         fclose( cfPtr );
27     }
28
29     return 0;
```

```

1 //Εγγραφή στο αρχείο «credit.dat» στοιχείων ταξινομημένα με βάση το
2
3 #include <stdio.h>
4
5 struct clientData {
6     int acctNum;
7     char lastName[ 15 ];
8     char firstName[ 10 ];
9     double balance;
10 };
11
12 int main()
13 {
14     FILE *cfPtr;
15     struct clientData client = { 0, "", "", 0.0 };
16
17     if ( ( cfPtr = fopen( "credit.dat", "r+" ) ) == NULL )
18         printf( "File could not be opened.\n" );
19     else {
20         printf( "Enter account number
21                 ( 1 to 100, 0 to end input )\n? " );
22         scanf( "%d", &client.acctNum );
23
24         while ( client.acctNum != 0 ) {
25             printf( "Enter lastname, firstname, balance\n? " );
26             fscanf( stdin, "%s%s%lf", client.lastName,
27                     client.firstName, &client.balance );
28             fseek( cfPtr, ( client.acctNum - 1 ) *
29                     sizeof( struct clientData ), SEEK_SET );
30             fwrite( &client, sizeof( struct clientData ), 1,
31                     cfPtr );
32             printf( "Enter account number\n? " );
33             scanf( "%d", &client.acctNum );
34         }
35         fclose( cfPtr );
36         return 0;
37     }
38 }
39

```

Enter account number (1 to 100, 0 to end input)

? 37

Enter lastname, firstname, balance

? Barker Doug 0.00

Enter account number

? 29

Enter lastname, firstname, balance

? Brown Nancy -24.54

Enter account number

? 96

Enter lastname, firstname, balance

? Stone Sam 34.98

Enter account number

? 88

Enter lastname, firstname, balance

? Smith Dave 258.34

Enter account number

? 33

Enter lastname, firstname, balance

? Dunn Stacey 314.33

Enter account number

? 0

