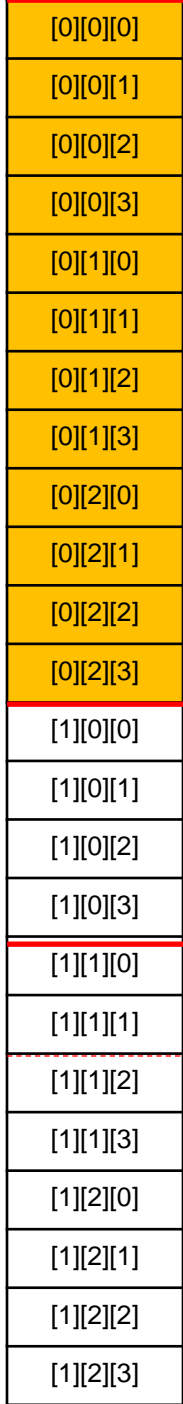


Διαδικαστικός Προγραμματισμός

Βασίλης Παλιουράς



```
int array3D[2][3][4];
           h  w  d

(int *) array3D + i*w*d + j * d + k
```

$$1*w*d = 1 * 3 * 4$$

$$1*d = 1 * 4$$

2

```
Το στοιχείο array3D[1][1][2]
βρίσκεται στη θέση
(int *) array3D + 1*3*4+ 1*4 + 2
```

Ευέλικτος κώδικας

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

```
void readarray(int *, int, int, int);
void printarray(int *, int, int, int);
void print3D(int [2][3][4]);
```

```
int main( ) {
```

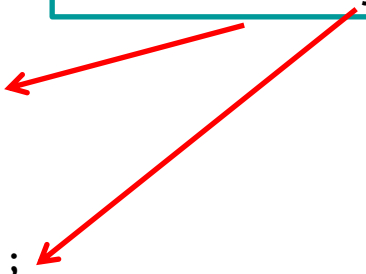
```
    int array3D[2][3][4];
    int another3D[2][2][2];
```

```
    readarray( (int *)array3D, 2, 3, 4);
    printarray((int *)array3D, 2, 3, 4);
    print3D(array3D);
```

```
    readarray( (int *)another3D, 2, 2, 2);
    printarray((int *)another3D, 2, 2, 2);
    return 0;
```

```
}
```

Ίδια συνάρτηση, η γεωμετρία ως του πίνακα ως παράμετρος.



```
#include <stdio.h>
#include <stdlib.h>
//
// C99 comment style
// Demonstration of C99 VLAs
//
void readarray(int rows, int cols, int x[rows][cols]);
void printarray(int rows, int cols, int x[rows][cols]);

int main( ) {
    int a[5][5];
    readarray(5, 5, a);
    printarray(3, 5, a);
    return 0;
}

void readarray(int rows, int cols, int x[rows][cols]) {
    int i, j;

    for (i = 0; i < rows; i++)
        for (j = 0; j < cols; j++)
            x[i][j] = -i*cols - j ;
}

void printarray(int rows, int cols, int x[rows][cols]) {
    int i, j;

    for (i = 0; i < rows; i++) {
        for (j = 0; j < cols; j++)
            printf("%4d", x[i][j]) ;
        printf("\n");
    }
}
```

Βασική διαχείριση αρχείων

- Πρότυπα στο `stdio.h`
- Τύπος `FILE`

```
FILE * stream;
```

- Χρήσιμες συναρτήσεις

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

```
int fclose(FILE * stream)
```

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

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

```
int feof(FILE * stream)
```

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

```
...
```

Επιπλέον στοιχεία σε παραδείγματα

- Macros από `stdlib.h`

`EXIT_SUCCESS` ($\rightarrow 0$)

`EXIT_FAILURE`

- Συναρτήσεις με πρότυπα στο `stdlib.h`

`int exit(int status)`

`int system(const char * string)`

- Compiler options:

`-std=c90 -Wall -Wextra -pedantic -Werror`

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,
 - δυναμική ανάγνωση,
 - δυναμική εγγραφή, ...
- Επεξεργαζόμαστε με κατάλληλες συναρτήσεις
- Κλείνουμε το αρχείο με κατάλληλο fclose()

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;  
}
```

Δείκτης σε FILE

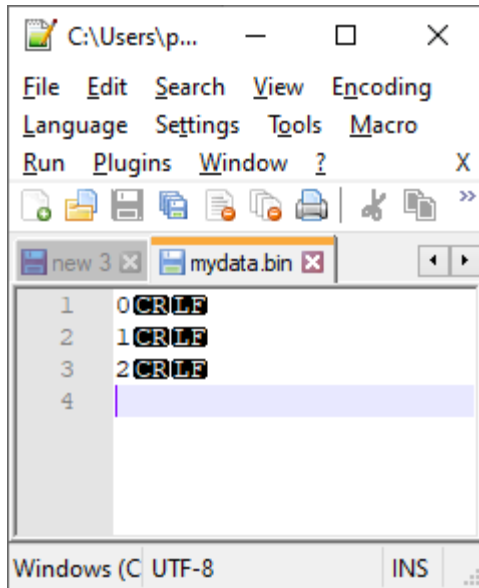
Όνομα αρχείου

Άνοιγμα αρχείου για
εγγραφή κειμένου
(ascii)

Αν το αρχείο δεν υπάρχει,
το δημιουργεί.

Αν υπάρχει, γράφει νέο
επάνω του...

- Παράδειγμα διαφοράς "wt" και "wb" σε windows

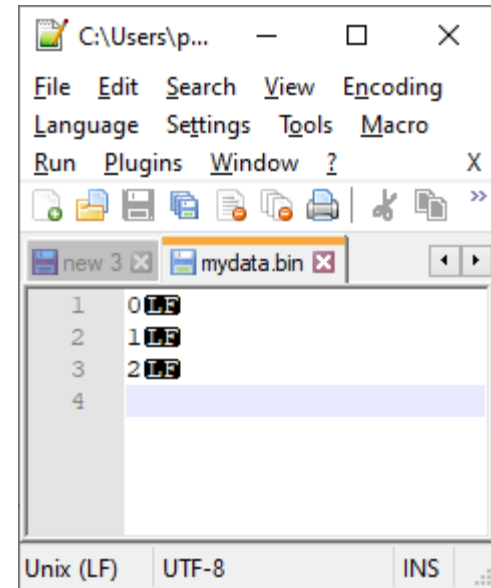


A screenshot of a Notepad window titled "C:\Users\p...". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. The toolbar shows icons for file operations. The window has two tabs: "new 3" and "mydata.bin". The text in the editor is:

```
1 0CRLF
2 1CRLF
3 2CRLF
4
```

The status bar at the bottom shows "Windows (C UTF-8" and "INS".

"wt", 9 bytes



A screenshot of a Notepad window titled "C:\Users\p...". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. The toolbar shows icons for file operations. The window has two tabs: "new 3" and "mydata.bin". The text in the editor is:

```
1 0LF
2 1LF
3 2LF
4
```

The status bar at the bottom shows "Unix (LF)" and "UTF-8".

"wb", 6 bytes

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

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

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

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

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

```
    fclose(myinput);
```

```
    return EXIT_SUCCESS;
```

```
}
```

Διαβάζω έναν αριθμό

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`

Ανάγνωση γνωστού πλήθους αριθμών

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

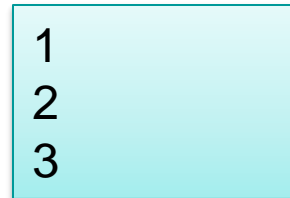
```
int main(void) {
    FILE *myinput;
    int d, i;

    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;
}
```



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

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

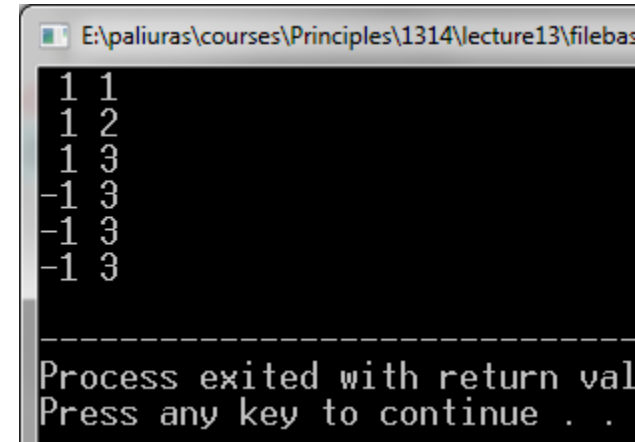
```
    for (i = 0; i < 2*N ; i++) {
        a = fscanf(myinput, "%d", &d);
        printf("%2d %d\n", a, d);
    }
```

```
    fclose(myinput);
```

```
    return EXIT_SUCCESS;
```

```
}
```

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



```
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 . . .
```

Λάθος χρήση του EOF

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

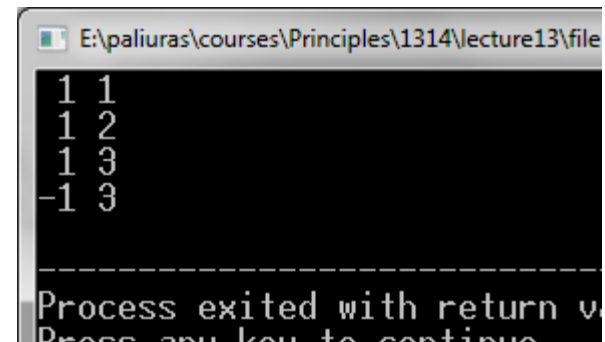
```
int main(void) {
    FILE *myinput;
    int d, a;

    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;
}
```



```
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

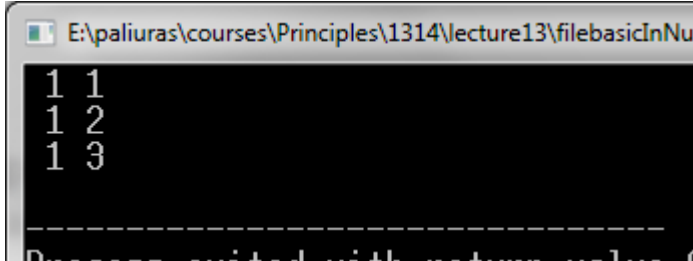
```
int main(void) {
    FILE *myinput;
    int d, a;

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

    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);
    }

    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 0

```



```
#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);
    }

    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;
}
```

```
#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;
}
```

Διαβάζω χαρακτήρες

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

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];

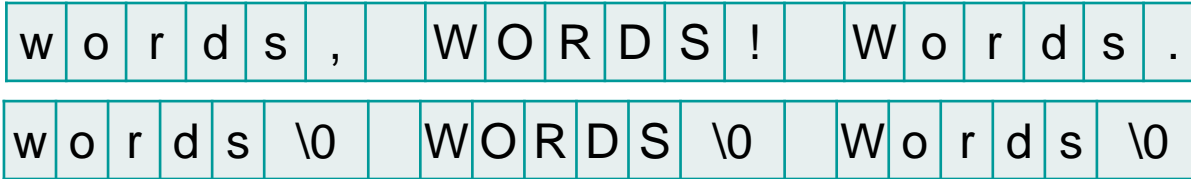
    if ( (textin = fopen("mytext.txt", "rt"))==NULL) {
        printf("cannot open file.\n");
        exit(EXIT_FAILURE);
    }

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

    fclose(textin);

    return EXIT_SUCCESS;
}
```

`char * strtok (char * str, const char * delim)`

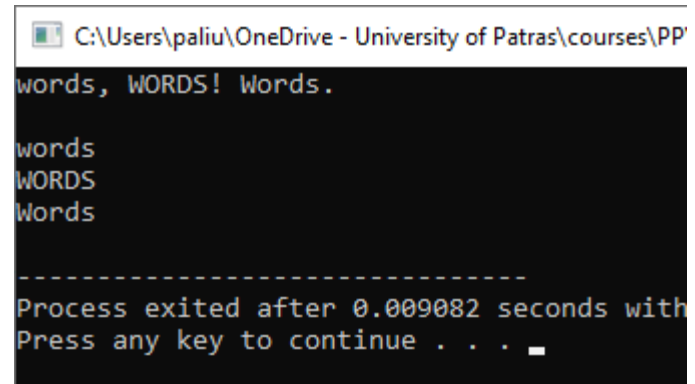


```
#include <stdio.h>
#include <string.h>
#define SIZE 20
int main(void) {
    int i;
    char str[] = "words, WORDS! Words.";

    char * word ;
    printf("%s\n\n", str);

    word = str;
    while ( (word = strtok(word, ",! ."))!=NULL) {
        printf("%s\n", word);
        word = NULL;
    }

    return 0;
}
```



```

#include <stdio.h>
#include <string.h>
#define SIZE 20
int main(void) {
    int i;
    char str[] = "words, WORDS! Words.";

    char * word ;
    printf("%s\n\n", str);

    word = str;
    while ( (word = strtok(word, ",! .")) != NULL) {
        printf("%s\n", word);
        word = NULL;
    }

    return 0;
}

```

```

C:\Users\paliu\OneDrive - University of Patras\courses\PP
words, WORDS! Words.
words
WORDS
Words
-----
Process exited after 0.009082 seconds with
Press any key to continue . . . _

```

```

#include <stdio.h>
#include <string.h>
#define SIZE 20
int main(void) {
    int i;
    char str[] = "words, WORDS! Words.";

    char * word ;
    printf("%s\n\n", str);

    for (word = str; word = strtok(word, ",! ."); word = NULL)
        printf("%s\n", word);

    return 0;
}

```

```

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

int main(void) {

    FILE * textin ;
    char word[100];

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

    while (fscanf(textin,"%s",word) != EOF) {
        printf("%s\n", word);
        {
            char * word_ptr ;
            char * delimiters = ", !.?" ;
            for (word_ptr = word; word_ptr = strtok(word_ptr, delimiters); word_ptr=NULL) {
                printf(">>>%s \n", word_ptr);
            }
        }
    }

    return EXIT_SUCCESS;
}

```



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

```
    fclose(textin);
```

```
    return EXIT_SUCCESS;
```

```
}
```

Re-factoring

```
void clearword(char * word) {
    char * word_ptr;
    char * delimiters = ", !.?";
    for (word_ptr = word; word_ptr = strtok(word_ptr, delimiters); word_ptr=NULL) {
        printf(">>>%s \n", word_ptr);
    }
    return ;
}
```

```
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;
```

```
}
```

```
#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;
}
```

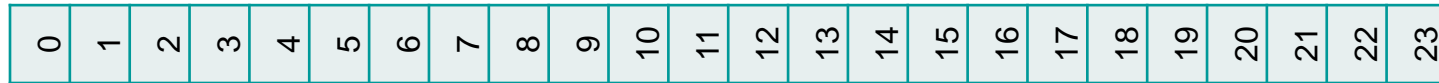
Πήγαινε 12 θέσεις από την
Αρχή του αρχείου

Δημιουργία αρχείου για εγγραφή

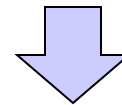
```
void createfile(void) {
    FILE *a;
    a = fopen("mytest.txt", "wt");
    fprintf(a, "my message: hello there!\n");
    fclose(a);
    return ;
}
```

```
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



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>
#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;
}
```

```
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

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

Επιστρέφει τη θέση στην οποία βρισκόμαστε στο τρέχον stream

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

```
void dosomething(FILE *);
```

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

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

    dosomething(p);

    fclose(p);

    return EXIT_SUCCESS;
}
```

Είσοδος στη συνάρτηση το p

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

```

#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

Βρείτε τις διαφορές

Κώδικας Α

```
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 Β

```
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 ;
}
```

```
#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;
}
```

By reference

FILE * → FILE **

```
#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 by value

```
#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 fileclose(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;
}
```

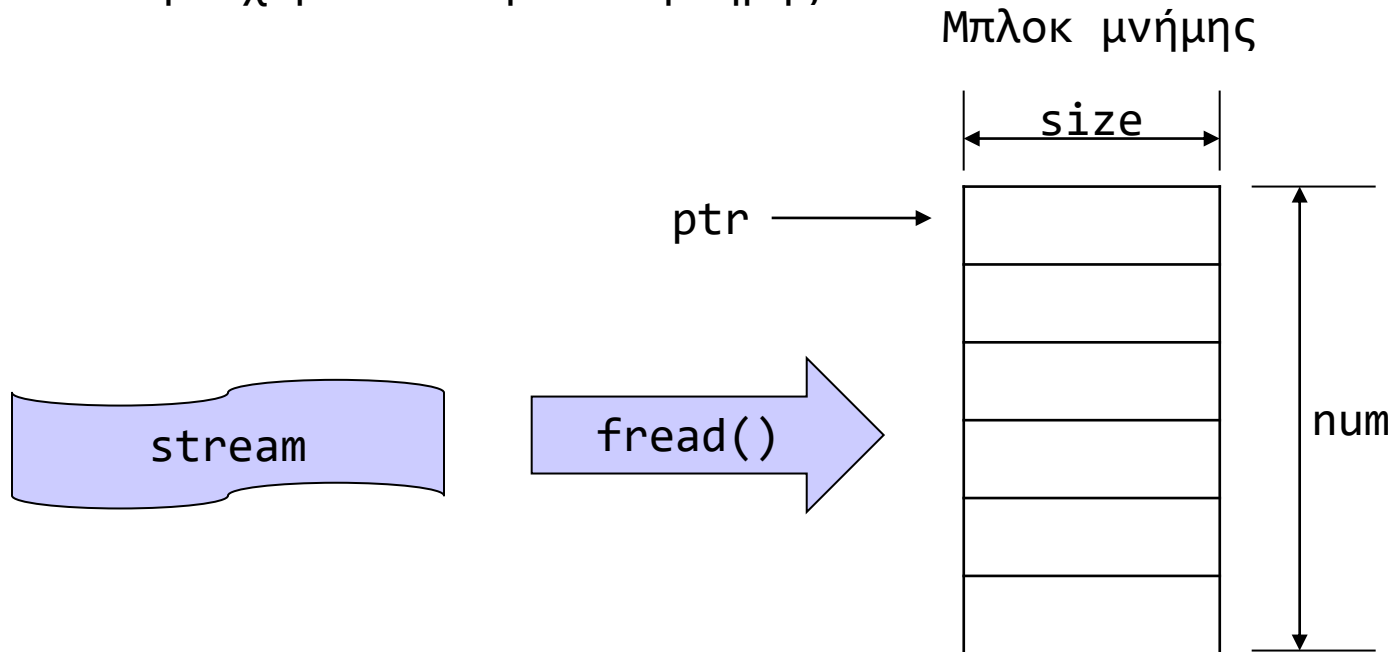
```
FILE * fopen(void) {
    FILE *p;

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

    return p;
}
```

```
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

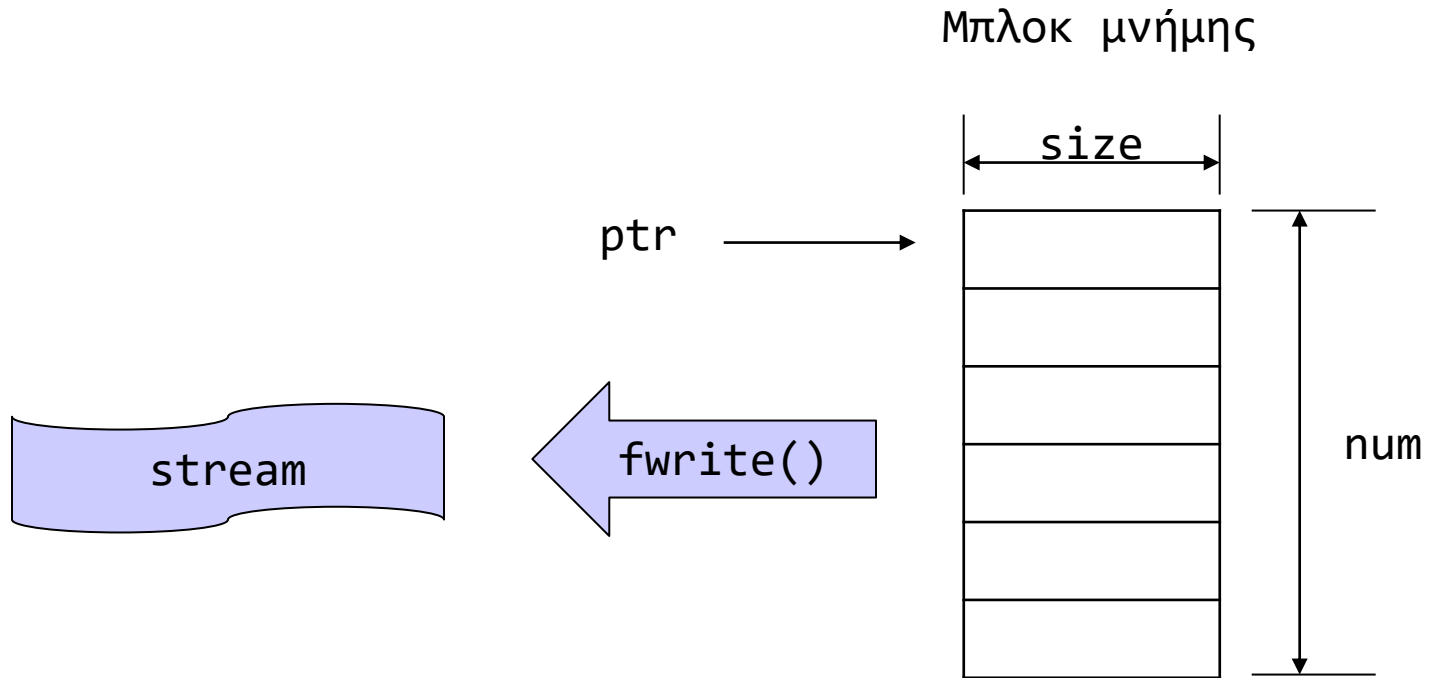
`size`: μέγεθος κάθε στοιχείου (σε bytes)

`num`: πλήθος στοιχείων

Επιστρέφει το πλήθος των στοιχείων που αναγνώστηκαν.
(αν διάφορο του `num`, έχει συμβεί λάθος ή EOF)

```
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 );
```

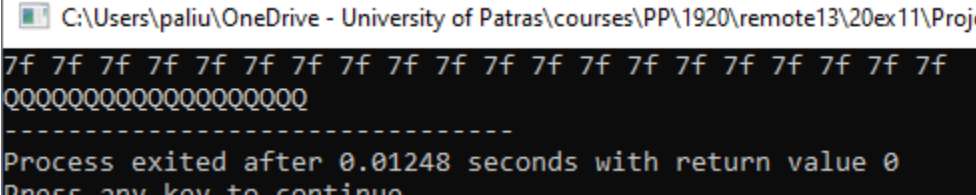
```
#include <stdio.h>
#include <string.h>
#define SIZE 20
int main(void) {
    int i;
    char bytes[SIZE];

    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
000000000000000000000000
-----
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;
```

```
}
```

C:\Users\paliu\OneDrive - University of Patras\courses\PP\1920\remote13\20ex09

```
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 . . .
```

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

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

    return;
}
```

Πού αποθηκεύονται παράμετροι και τοπικές μεταβλητές: stack

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

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

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

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

```
    g(1);
```

```
    h(1.0);
```

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

```
    f(2);
```

```
    g(1);
```

```
    return 0;
```

```
}
```

Η 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;  
}
```

```
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;
}
```

Εικόνα του stack κατά τη διάρκεια εκτέλεσης των συναρτήσεων

```

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 . . .

```

Κάθε φορά διατίθεται ο ίδιος χώρος μνήμης

