

Ορίσματα στη συνάρτηση main

```
#include <iostream>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])  
{  
    cout << "argc = " << argc << endl;  
    for( int i=0 ; i < argc ; i++ )  
        cout << "argv[" << i << "] = " << argv[i] << endl;  
    return 0;  
}
```

Παράδειγμα

```
#include <iostream>
using namespace std;

int main( int argc, char* argv[] )
{
    cout << "The name used to start the program: " << argv[ 0 ]
         << "\nArguments are:\n";
    for (int n = 1; n < argc; n++)    cout << n << ": " << argv[ n ] << '\n';
    return 0;
}
```

> a Hello world!

The name used to start the program: a Arguments are: 1: Hello 2: world!

> a.exe "Peter Piper" picked a peck of "pickled peppers"

The name used to start the program: test\a.exe Arguments are: 1: Peter Piper 2: picked 3: a 4: peck 5: of 6: pickled peppers

Διαχείριση Αρχείων

- Τα αρχεία δημιουργούνται σε κάποιο αποθηκευτικό μέσο και σε αυτά αποθηκεύουμε μόνιμα κάποιες πληροφορίες.
- Υπάρχουν 2 είδη αρχείων.
 - Αυτά στα οποία αποθηκεύουμε χαρακτήρες (ASCII) και
 - αυτά στα οποία αποθηκεύουμε εγγραφές (binary).

Διαχείριση Αρχείων σε Υψηλό Επίπεδο

Άνοιγμα αρχείου

```
FILE *δείκτης;
```

```
δείκτης=fopen("όνομα αρχείου","w");
```

- Π.χ.

```
FILE *fp;
```

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

– Το αρχείο new.txt δημιουργείται στον τρέχοντα κατάλογο

Άνοιγμα αρχείου

```
δείκτης=fopen("όνομα αρχείου","r");
```

- Π.Χ.

```
FILE *fp
```

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

Κλείσιμο αρχείου

`fclose` (όνομα δείκτη)

- Π.χ.
`fclose(fp);`
- Σημείωση : Το αρχείο ανοίγει είτε για διάβασμα είτε για γράψιμο(αποθήκευση).
- Αν θέλουμε σε κάποιο πρόγραμμα να ανοίξουμε ένα αρχείο τόσο για διάβασμα όσο και για γράψιμο τότε πρέπει αφού ολοκληρώσουμε την μία διαδικασία να το κλείσουμε και πριν το ανοίξουμε για την άλλη διαδικασία.

Διάβασμα από αρχείο

fscanf(δείκτης, format, &arg1, &arg2, ..., &argm);

- π.χ.

```
int a;
```

```
fscanf(fp, "%d", &a);
```

- Σημείωση : Για να πραγματοποιήσουμε διάβασμα πρώτα το έχουμε ανοίξει με το πρόθεμα "r".

Γράψιμο σε αρχείο

`fprintf(δείκτης, format, agr1, arg2, ..., argm);`

- Π.χ.
`fprintf(fp, "%dc\n", a);`
- Σημείωση : Για να πραγματοποιήσουμε αποθήκευση πρώτα το έχουμε ανοίξει με το πρόθεμα "w".

Επέκταση αρχείου (append)

```
δείκτης=fopen("όνομα αρχείου","a");
```

- Π.χ.
FILE *fp;
fp=fopen("new","a");

Παράδειγμα

- Ακολουθεί ένα παράδειγμα, όπου γίνεται χρήση αρχείου, του οποίου το όνομα καθορίζει ο Χρήστης στη διάρκεια της εκτέλεσης του προγράμματος, όπου φορτώνονται ονόματα, διευθύνσεις και τηλέφωνα:

```
// ενημέρωση τηλεφώνων
```

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

```
void main ( )
```

```
{ FILE *f;
```

```
char nameout[20], name[31],  
address[41], telephon[20];
```

```
printf("Name of file :");
```

```
scanf("%s", nameout);
```

```
f = fopen (nameout,"a");//open existing
```

```
if (f == NULL) f = fopen(nameout, "w");
```

```
printf("GIVE NEW DATA(if a name=*  
then EXIT):\n");
```

```
do
```

```
{ printf("name (until 30  
characters):");
```

```
scanf("%s", name);
```

```
if(name[0] == '*') break;
```

```
printf("address (until 40  
characters):");
```

```
scanf("%s", address);
```

```
printf("telephone number:");
```

```
scanf("%s", telephon);
```

```
fprintf(f,"%s %s %s\n", name,  
address, telephon);
```

```
}
```

```
while(true);
```

```
fclose(f);
```

```
}
```

Μια τυπική έξοδος του προγράμματος είναι η ακόλουθη όπου ο Χρήστης έδωσε για πρώτη φορά δυο ονόματα.

Name of file :mytel.txt

GIVE NEW DATA(if a name=* then EXIT):

name (until 30 characters):name001

address (until 40 characters):addr001

telephone number:tel001

name (until 30 characters):name002

address (until 40 characters):addr002

telephone number:tel002

name (until 30 characters):*

οπότε το περιεχόμενο του αρχείου mytel.txt θα είναι το ακόλουθο:

name001 addr001 tel001

name002 addr002 tel002

Διαχείριση Αρχείων σε Υψηλό Επίπεδο με τη Βιβλιοθήκη `fstream`

- `<fstream.h>` (`<fstream>`): Αρχείο-επικεφαλίδα που πρέπει να περιλάβουμε αν θέλουμε να κάνουμε είσοδο ή έξοδο από και προς αρχεία.
- Ορίζουν διάφορες κλάσεις μεταξύ των οποίων και:
 - `ifstream` για είσοδο
 - `ofstream` για έξοδο
 - `fstream` για είσοδο / έξοδο

Παράδειγμα

```
// basic file operations
#include <iostream>
#include <fstream>
using namespace std;

int main () {
    ofstream myfile;
    myfile.open ("example.txt");
    myfile << "Writing this to a file.\n";
    myfile.close();
    return 0;
}
```

Άνοιγμα αρχείου

`open (filename, mode)` (το `mode` είναι προαιρετικό)

<code>ios::in</code>	Open for input operations
<code>ios::out</code>	Open for output operations
<code>ios::binary</code>	Open in binary mode
<code>ios::ate</code>	Set the initial position at the end of the file
<code>ios::app</code>	All output operations are performed at the end of the file, appending the content to the current content of the file
<code>ios::trunc</code>	If the file is opened for output operations and it already existed, its previous content is deleted and replaced by the new one

```
ofstream myfile;  
myfile.open ("example.bin", ios::out | ios::app | ios::binary);
```

ή

```
ofstream myfile ("example.bin", ios::out | ios::app | ios::binary);
```


Κλείσιμο αρχείου

- When we are finished with our input and output operations on a file we shall close it so that the operating system is notified and its resources become available again.
- For that, we call the stream's member function `close`. This member function takes flushes the associated buffers and closes the file:

```
myfile.close();
```

Έξοδος μεταβλητών σε αρχείο

```
#include <iostream>
#include <fstream>
using namespace std;

int main()
{
    ofstream out("test");
    if (!out) {
        cout << "Cannot open file.\n";
        return 1;
    }

    out << 10 << " " << 123.23 << "\n";
    out << "This is a short text file.";

    out.close();
    return 0;
}
```

Είσοδος μεταβλητών από αρχείο

```
#include <iostream>
#include <fstream>
using namespace std;

int main()
{
    char ch; int i; float f; char str[80];

    ifstream in("test");
    if(!in) {
        cout << "Cannot open file.\n";
        return 1;
    }
    in >> i >> f >> ch >> str;

    cout << i << " " << f << " " << ch << endl;
    cout << str << endl;

    in.close();
    return 0;
}
```

Μέθοδοι get και put

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

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

```
{
```

```
    if (argc!=3) cout << "wrong number of arguments"<<endl;
```

```
    ifstream from(argv[1]);
```

```
    ofstream to(argv[2]);
```

```
    char ch;
```

```
    while (from.get(ch)) to.put(ch);
```

```
    if (!from.eof() ||!to) cout<<"something strange happened"<<endl;
```

```
    to.close();
```

```
    from.close();
```

```
    return 0;
```

```
}
```