

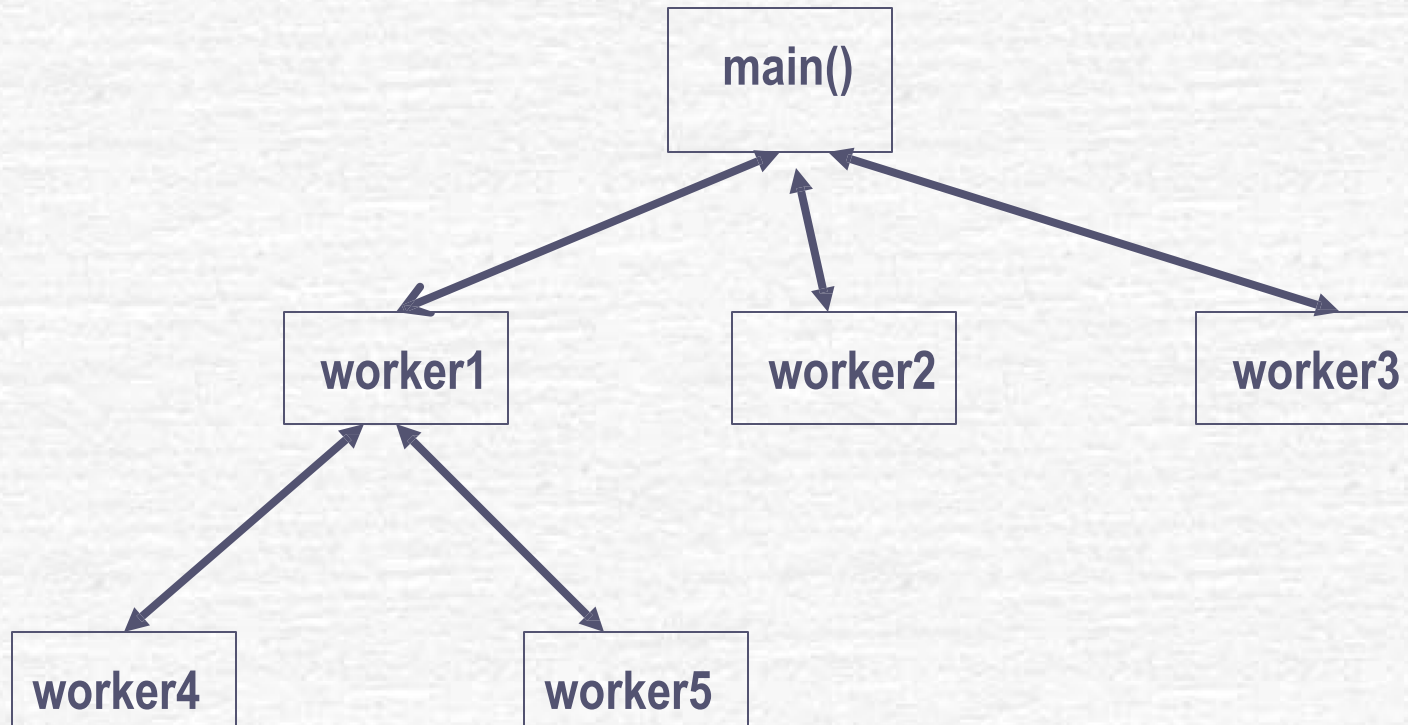


**Τμήμα Μηχανικών Η/Υ & Πληροφορικής
Πανεπιστήμιο Πατρών**

Εισαγωγή στον Προγραμματισμό Υπολογιστών

Η γλώσσα προγραμματισμού C

Συναρτήσεις (Functions)



Υποπρογράμματα

- Αποτελούν τον τρόπο εφαρμογής (υλοποίησης) της τμηματοποίησης σε ένα πρόγραμμα.
- Κάθε υποπρόγραμμα είναι ένα αυτόνομο τμήμα (μικρό πρόγραμμα)
- Ο συνδυασμός των υποπρογραμμάτων συνιστά το (αρχικό) πρόγραμμα.
- Πλεονεκτήματα
 - Αποφυγή επαναλήψεων
 - Αύξηση επαναχρησιμοποίησης
 - Βελτίωση αναγνωσιμότητας
 - Ευκολότερη συντήρηση

Συναρτήσεις στη C

- Πρόγραμμα C = σύνολο συναρτήσεων
- Η συνάρτηση `main` αντιπροσωπεύει το κυρίως πρόγραμμα, δηλ. τον τρόπο με τον οποίο συνδυάζονται οι υπόλοιπες συναρτήσεις για τη λύση του προβλήματος.
- Μέρη συνάρτησης
 - Κεφαλίδα = η διεπαφή της συνάρτησης (όνομα, είσοδος, έξοδος)
 - Σώμα = υλοποίηση/ορισμός της συνάρτησης
- Προτάσεις συνάρτησης
 - Δήλωση – Κλήση - Ορισμός

Δήλωση Συνάρτησης

- Προσδιορίζεται η διεπαφή, ο τρόπος αναφοράς στη συνάρτηση:

<τύπος> <όνομα – συν> ([<παράμετροι>]); Όπου

<παράμετροι>:=<τύπος1> [<όνομ-παρ1>], ..., <τύποςN> [<όνομ-παρN>]

↓
τυπικά ορίσματα (ή είσοδοι)

→ τυπικά αποτελέσματος (εξόδου)

- Αν η συνάρτηση δεν επιστρέφει κάποια τιμή, τότε χρησιμοποιείται σαν τύπος αποτελέσματος η λέξη κλειδί void.

Παραδείγματα

```
int max (int a, int b);
```

```
int min (int, int);
```

```
double exp ( double m, int n);
```

```
void swap (int a, int b);
```

```
char *getptr (char * str, char ch);
```

```
char *getptr (char str [ ], char ch);
```

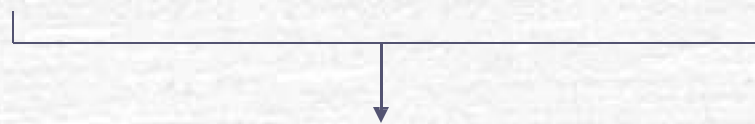
Παράδειγμα

```
#include <stdio.h>
int square(int y); /* αρχέτυπο συνάρτησης */
int main ( )
{
    int x;
    for (x=1; x<=10; x++)
        printf("%d", square(x));
    printf("\n");
    return 0;
}
/* ορισμός συνάρτησης */
int square( int y)
{
    return y * y;
}
```

Κλήση Συνάρτησης

- Καλείται η συνάρτηση για εκτέλεση με συγκεκριμένα ορίσματα

$\langle \text{όνομα} - \text{συν} \rangle (\langle \text{ορισ1} \rangle, \langle \text{ορισ2} \rangle, \dots, \langle \text{ορισN} \rangle);$



πραγματικά ορίσματα (σταθερές, μεταβλητές, εκφράσεις|)

- Τα πραγματικά ορίσματα πρέπει να είναι του ίδιου αριθμού και τύπου με τα τυπικά ορίσματα

Παραδείγματα

```
swap ( x, y);
```

```
draw_circle (a/2.0, 2.0 * b, c);
```

```
max_num = max(num1, num2);
```

```
min_num = (num, 5);
```

```
x = y + max (num1/2.0, 3.0*num2);
```

```
printf ( “ο μέγιστος είναι: %d\n”, max(x1,x2));
```

Ορισμός Συνάρτησης

- Για κάθε μη ενσωματωμένη (δική μας) συνάρτηση πρέπει να ορίσουμε το σώμα της στο πρόγραμμα (μετά την main)

```
<τύπος> (<όνομα - συν> (<παράμετροι>)
```

```
{
```

```
    <δηλώσεις τοπικών μεταβλητών>
```

```
    <προτάσεις>
```

```
}
```

Παράδειγμα (1)

```
double embadon (double platos, double mikos)
```

```
{
```

→ τυπικές παράμετροι

```
double apotelesma;
```

```
apotelesma = platos * mikos;
```

```
platos=15;
```

```
return (apotelesma);
```

Επιστροφή ελέγχου και
τιμής στην καλούσα συνάρτηση

```
}
```

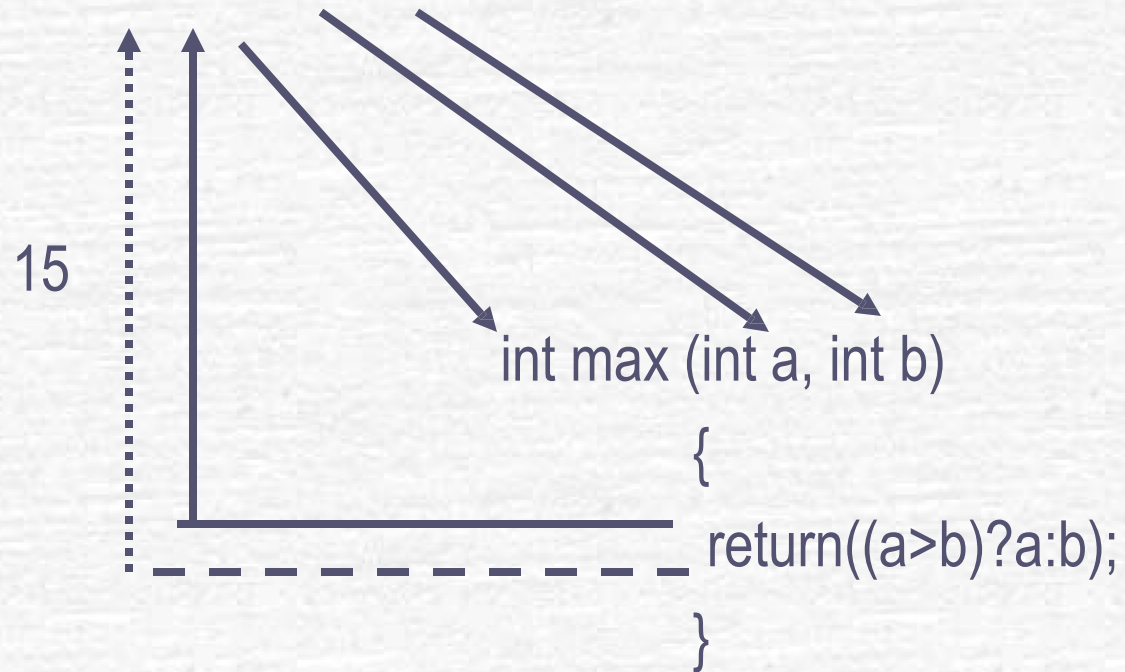
Παράδειγμα (2)

```
int max (int a, int b )  
{  
    int max;  
    max = ( a > b ) ? a : b;  
    return (max);  
}
```

```
int max( int a, int b)  
{  
    return ((a > b) ? a : b);  
}
```

Μηχανισμός Κλήσης (1)

```
max_num = max (max ( 15, 13), 17);
```



Αναδρομή (Recursion) (1)

- Ορισμός συνάρτησης μέσω κλήσης του εαυτού της
- Μία τεχνική επίλυσης προβλημάτων
- Π.χ. εύρεση αθροίσματος $1+2+\dots+n$

- Κλασσική λύση

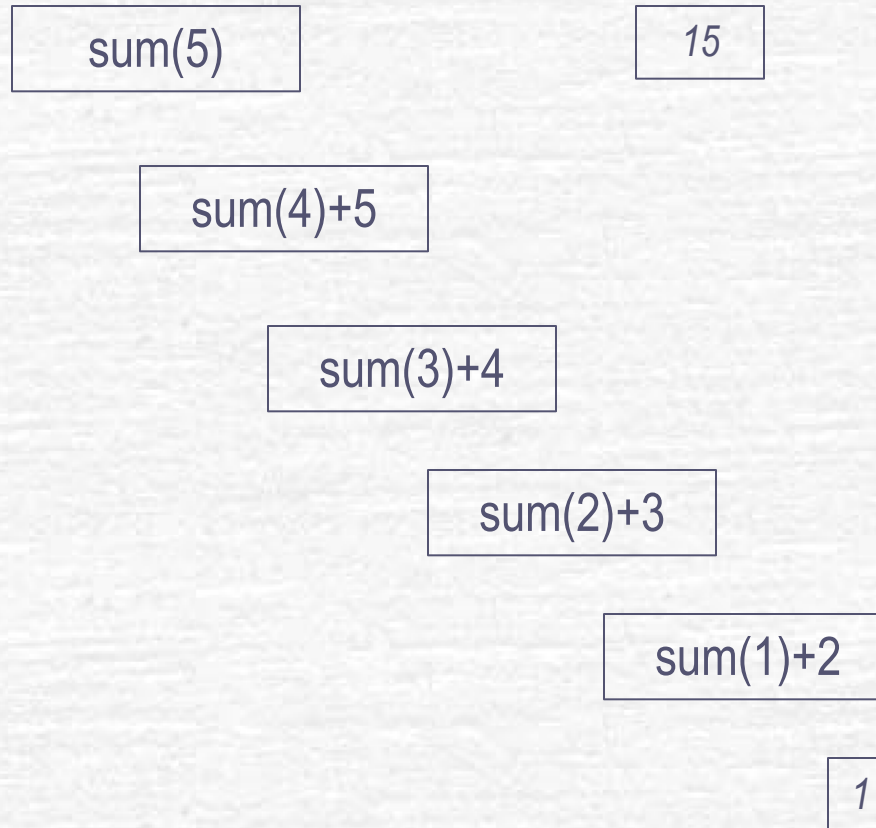
```
int sum (int n) {  
    s=0;  
    for (i=1; i<=n; i++)  
        s=s+i;  
    return s;}
```

Αναδρομή (Recursion) (2)

- Αναδρομική λύση

$$\text{sum}(n) = \text{sum}(n-1) + n$$

```
int sum (int n) {  
  if (n<=1)  
    return n;  
  else  
    return (sum(n-1)+n);}
```



Η συνάρτηση main()

Η συνάρτηση από την οποία ξεκινά η εκτέλεση του προγράμματος.

argc: Ο αριθμός των παραμέτρων που περνάμε στην main όταν εκτελούμε το πρόγραμμα

argv: πίνακας δεικτών στα αλφαριθμητικά της γραμμής διαταγών (command line) με την οποία εκτελούμε το πρόγραμμα.

HelloWorld.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(int argc, char *argv[]) {
5     int i;
6
7     printf("Hello world!\n");
8     printf("argc=%d\n", argc);
9     for(i=0; i<argc; i++)
10         printf("argv[%d]=%s\n", i, argv[i]);
11     return 0;
12 }
```

```
C:\Code\courses\I2P22-23\HelloWorld>helloworld "by kleanthis" Thramboulidis
```

```
Hello world!
```

```
argc=3
```

```
argv[0]=helloworld
```

```
argv[1]=by kleanthis
```

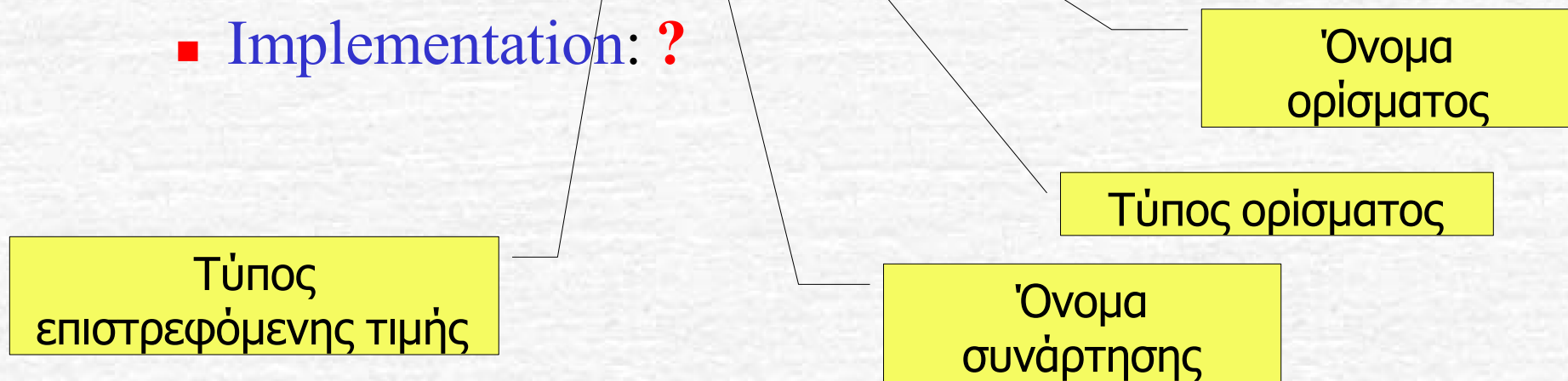
```
argv[2]=Thramboulidis
```

main(): Η συνάρτηση η οποία περιγράφει τη διεργασία που θα εκτελέσει ο υπολογιστής όταν τρέξουμε το πρόγραμμα.

Συνάρτηση - Ορισμός

Που είναι το interface και το implementation της συνάρτησης sqrt;

- είναι μία αυτόνομη μονάδα κώδικα σχεδιασμένη να επιτελεί συγκεκριμένο έργο.
- Π.χ. Έργο: Υπολογισμός τετραγωνικής ρίζας
 - Interface: **double sqrt(double x)**
 - Implementation: ?



Βασική Βιβλιοθήκη της C

- Ορίζει ένα σύνολο από συναρτήσεις που αναπαριστούν **βασικές διεργασίες**
- Η γνώση των συναρτήσεων αυτών αποτελεί **βασική προϋπόθεση** για την ανάπτυξη σύνθετων προγραμμάτων

Standard C library

<https://www.csse.uwa.edu.au/programming/ansic-library.html>

Standard C Library Functions Table, By Name - IBM

<https://www.ibm.com/docs/en/i/7.3?topic=extensions-standard-c-library-functions-table-by-name>

1. ANS X3.159-1989 πρότυπο

τον Δεκέμβριο του 1989 και δημοσιεύθηκε άνοιξη του 1990

2. ISO 9899-1990 πρότυπο

είναι σχεδόν αντίγραφο του ANSI. (ISO:International Organization for Standardization)

3. ISO/IEC 899:1999, το τρέχον πρότυπο

αντικατέστησε το 1990 ISO πρότυπο

Βασικές κατηγορίες συναρτήσεων της standard library

- εισόδου εξόδου
- μαθηματικές
- διαχείρισης αλφαριθμητικών
- ταξινόμηση χαρακτήρων και μετατροπές χαρακτήρων
- μετατροπής δεδομένων
- αναζήτησης και ταξινόμησης
- διαχείρισης αρχείων
- διαχείρισης μνήμης

Δες παράρτημα Γ
του βιβλίου

Συναρτήσεις I/O της BB

■ Εισόδου

```
int getchar(void) ;
```

```
int scanf(const char* format, ...);
```

■ Εξόδου

```
int putchar(int c);
```

```
int puts(const char* s);
```

```
int printf(const char *format,  
όρισμα2, όρισμα3,...);
```

Δες άσκηση 1 Κεφάλαιο 11
Χρήση συναρτήσεων BB

Δες άσκηση 4 Κεφάλαιο 11
Για προχωρημένα θέματα συναρτήσεων

Συνάρτηση της BB

Η συνάρτηση printf

```
#include <stdio.h> /* αρχείο επικεφαλίδας */
```

Το **πρωτότυπο** της (function prototype) είναι:

```
int printf(const char *format, όρισμα2,...) ;
```

- Το πρώτο όρισμα ορίζει σύμφωνα με δεδομένους κανόνες (δες εγχειρίδιο βασικής βιβλιοθήκης) τον αριθμό και τον τύπο των ορισμάτων που θα εξάγει η συνάρτηση στην κύρια έξοδο. Αυτά θα πρέπει να είναι σε συμφωνία με τα υπόλοιπα ορίσματα που ακολουθούν. Τις τιμές αυτών των ορισμάτων βγάζει η συνάρτηση στην έξοδο.

Χρήση συνάρτησης της BB -1

```
#include <stdio.h>
main()
{
    int num1 = 2;
    int num2 = 4;
    int sum;
    sum = 2 + 4;
    printf("2 + 4 = %d\n", sum);
    printf("%d + %d = %d\n", num1, num2, sum);
}
```

Χρήση συνάρτησης της BB -2

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

```
main()
```

```
{
```

```
double num1 = 16;
```

```
double result;
```

```
result = sqrt(num1);
```

```
printf("sqrt of %f is %f\n", num1, result);
```

```
printf("sqrt of %f is %f\n", num1, sqrt(num1) );
```

```
}
```

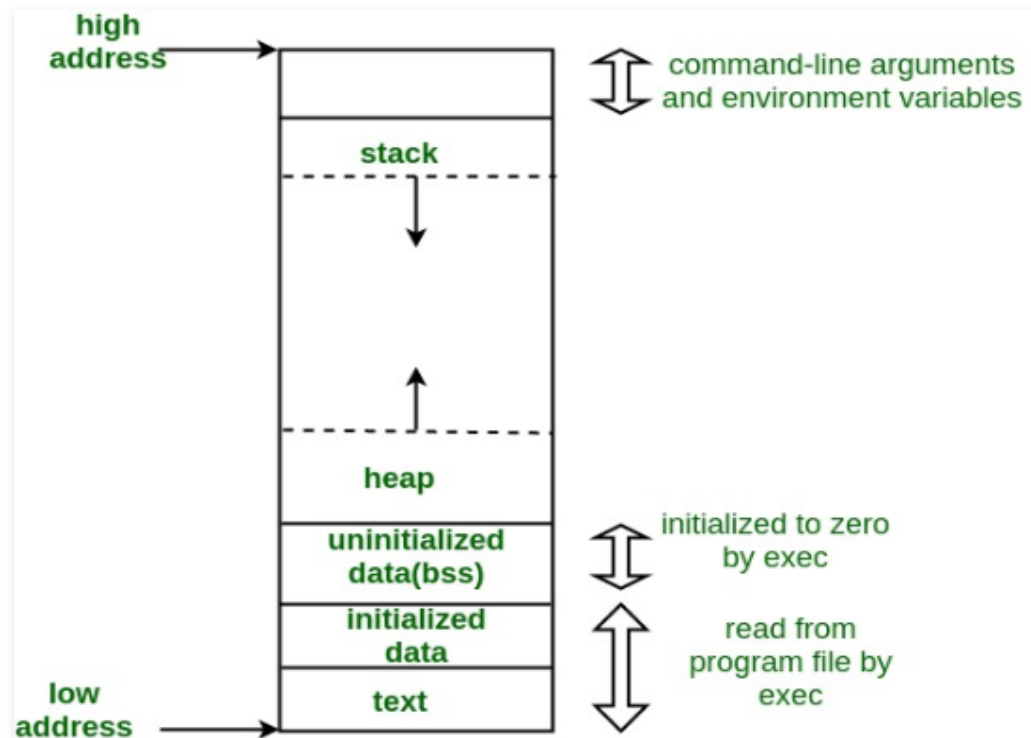
#include <math.h>

Αξιοποίηση
επιστρεφόμενης τιμής

Memory Layout of C Programs

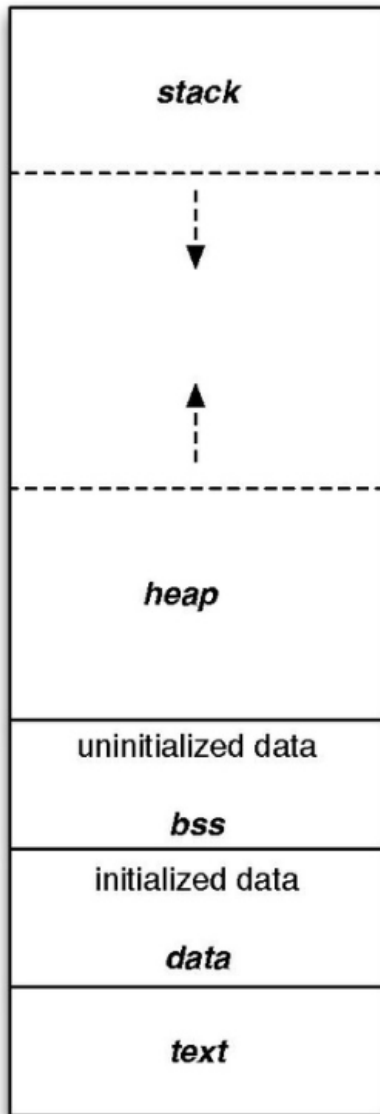
A typical memory representation of C program consists of following sections.

1. Text segment
2. Initialized data segment
3. Uninitialized data segment
4. Stack
5. Heap



A typical memory layout of a running process

Memory Layout of a C program



https://commons.wikimedia.org/wiki/File:Program_memory_layout.pdf

This file is licensed under the [Creative Commons Attribution-Share Alike 3.0 Unported](https://creativecommons.org/licenses/by-sa/3.0/) license.

Εμβέλεια Μεταβλητών

- Εμβέλεια = το τμήμα του προγράμματος στο οποίο έχει ισχύ ή είναι ορατή η μεταβλητή.
- Διάρκεια ζωής= πόσο υπάρχει στη μνήμη.
- Πότε μας αφορά η εμβέλεια;
Όταν έχουμε μεταβλητές με το ίδιο όνομα.

Τύποι Εμβέλειας (1)

- Εξωτερικές μεταβλητές

■ Γενικές ή καθολικές μεταβλητές (global variables)

- Δηλώνονται έξω και πάνω από κάθε συνάρτηση (και την main), στο τμήμα δηλώσεων μεταβλητών.
- Η εμβέλειά τους είναι όλο το αρχείο του πηγαίου κώδικα που

ανήκει η δήλωση, από το σημείο της δήλωσης και κάτω (εμβέλεια αρχείου)

Τύποι Εμβέλειας (2)

- Εσωτερικές ή αυτόματες μεταβλητές
 - Τοπικές μεταβλητές (local variables)
 - Δηλώνονται μέσα σε μία συνάρτηση.
 - Η εμβέλειά τους είναι το σώμα της συνάρτησης (εμβέλεια μπλοκ)
 - Τυπικές παράμετροι συναρτήσεων
 - Η εμβέλειά τους είναι το σώμα της συνάρτησης (εμβέλεια μπλοκ)
 - Μεταβλητές σύνθετης πρότασης
 - Δηλώνονται μέσα σε μία σύνθετη πρόταση
 - Η εμβέλειά τους είναι το σημείο της δήλωσης μέχρι το τέλος της σύνθετης πρότασης (εμβέλεια μπλοκ).
 - Στατικές Μεταβλητές
 - Δηλώνονται σε κάποια (ες) συνάρτηση (εις) (μετά την main) με τη λέξη κλειδί static πριν από τον τύπο μεταβλητής.
 - Η εμβέλειά τους είναι το αρχείο του πηγαίου κώδικα που ανήκει η δήλωση, από το σημείο της δήλωσης και κάτω (εμβέλεια αρχείου)

Κανόνες Εμβέλειας Μεταβλητών

- Μεταβλητές με το ίδιο όνομα επιτρέπονται μόνο όταν έχουν διαφορετική εμβέλεια
- Μεταβλητή με μικρότερη εμβέλεια αποκρύπτει πιθανώς ομώνυμες μεταβλητές μεγαλύτερης εμβέλειας.

Εμβέλεια Συνάρτησης

- Οι συναρτήσεις, όπως και οι μεταβλητές έχουν εμβέλεια
- Η εμβέλεια μίας συνάρτησης εκτείνεται από το σημείο της δήλωσής της μέχρι το τέλος του προγράμματος.
- Αν μία συνάρτηση δηλωθεί static, τότε η εμβέλειά της περιορίζεται το αρχείο που δηλώθηκε.
- Μεταβλητή με μικρότερη εμβέλεια αποκρύπτει πιθανώς ομώνυμες μεταβλητές μεγαλύτερης εμβέλειας.

Διάρκεια Μεταβλητής

- Ο χρόνος δέσμευσης της μνήμης που περιέχει την τιμή της μεταβλητής
- Καθολική μεταβλητή: διάρκεια εκτέλεσης προγράμματος (πλήρης διάρκεια).
- Τοπική μεταβλητή: διάρκεια εκτέλεσης συνάρτησης (περιορισμένη διάρκεια)
- Τυπική παράμετρος: διάρκεια εκτέλεσης συνάρτησης (περιορισμένη διάρκεια)
- Στατική τοπική μεταβλητή: διάρκεια εκτέλεσης προγράμματος (πλήρης διάρκεια)

Αρχικοποίηση Μεταβλητών

- Μία τοπική μεταβλητή περιορισμένης διάρκειας αρχικοποιείται με κάθε είσοδο στο μπλοκ (συνάρτηση) που ορίζεται
- Μία τοπική μεταβλητή πλήρους διάρκειας αρχικοποιείται με την έναρξη εκτέλεσης του προγράμματος
- Π.χ.

```
static int num;  
  
func (int) {  
    static int count = 0;  
    int num = 100; ... }
```

Παράδειγμα

```
#include <stdio.h>
void increment(void)
```

```
main() {
int j=0;
increment ( );
increment ( );
increment ( );
printf(“%d”, j}
```

```
void increment(void) {
int j=2;
static int k=2;
printf(“j: %d\t k: %d\n”, j++, k++); }
```

Πέρασμα Παραμέτρων

- Κατ' αξία ή τιμή (by value)

Η συνάρτηση δουλεύει σε αντίγραφα των πραγματικών παραμέτρων

- Κατ' αναφορά (by reference)

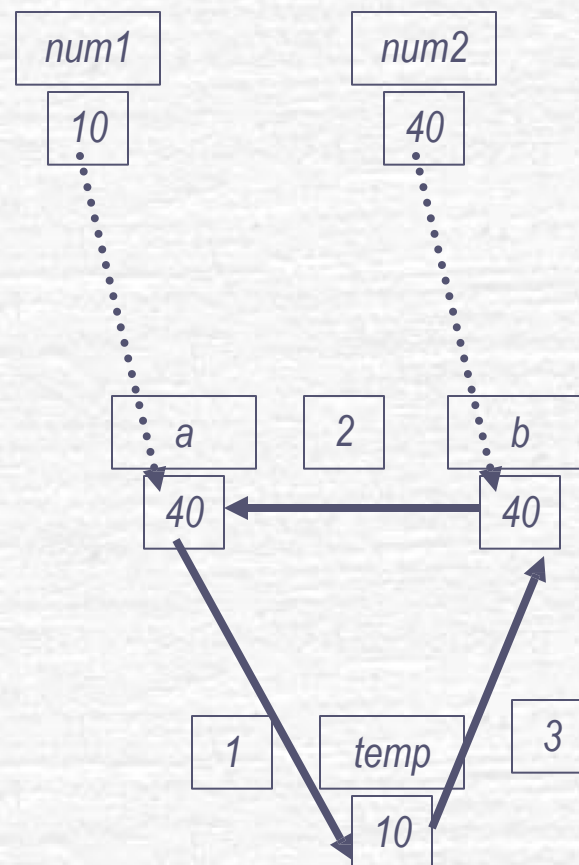
Η συνάρτηση δουλεύει στις πραγματικές παραμέτρους (μόνο για πίνακες)

Παράδειγμα (1)

```
void swap (int a, int b)
{ int temp;
  temp=a;
  a=b;
  b=temp;}
```

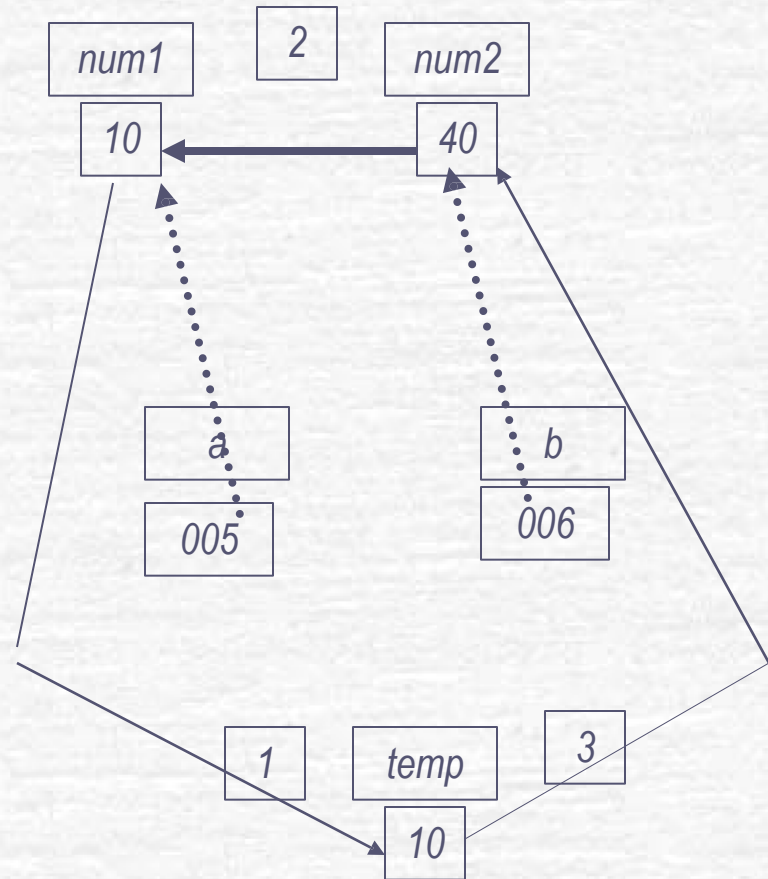
swap (num1, num2);

Ενώ οι τιμές των a,b αλλάζουν,
οι τιμές των num1, num2
παραμένουν αμετάβλητες



Παράδειγμα (2)

```
void swap (int *a, int * b)  
{ int temp;  
  temp=* a;  
  *a=*b;  
  *b=temp;}  
  
swap (&num1, &num2);
```



Πολλαπλά αρχεία

- Σε πολλαπλά αρχεία υπάρχει ΜΟΝΟ ένα αρχείο με τη `main()`
- Μία εξωτερική μεταβλητή μπορεί να χρησιμοποιηθεί από περισσότερα από ένα αρχεία αν και μόνο εάν η δήλωσή της στα άλλα αρχεία συνοδεύεται από `keyword extern`
- Μία συνάρτηση θα πρέπει να δηλωθεί σε κάθε αρχείο που τη χρησιμοποιεί, ορίζεται όμως ΜΟΝΟ σε ένα
- Αν μία συνάρτηση ή μεταβλητή δηλωθεί `static`, τότε η εμβέλειά της περιορίζεται αναγκαστικά στο αρχείο που δηλώθηκε.

Οδηγίες (<http://stackoverflow.com/questions/1433204/how-do-i-use-extern-to-share-variables-between-source-files-in-c>)

- Ένα header file περιέχει μόνο extern δηλώσεις μεταβλητών — ποτέ static
- Για κάθε μεταβλητή μόνο ένα header file τη δηλώνει (SPOT — Single Point of Truth). Αυτό το αρχείο κεφαλίδα συμπεριλαμβάνεται από το ΜΟΝΑΔΙΚΟ ΑΡΧΕΙΟ ΠΟΥ ΟΡΙΖΕΙ ΤΗ ΜΕΤΑΒΛΗΤΗ ΚΑΙ ΑΠΟ ΚΑΘΕ ΑΡΧΕΙΟ ΠΟΥ ΤΗΝ ΑΝΑΦΕΡΕΙ. Συνεπώς για κάθε μεταβλητή ΕΝΑ ΚΑΙ ΜΟΝΟ ΕΝΑ ΑΡΧΕΙΟ ΚΕΦΑΛΙΔΑ ΤΗ ΔΗΛΩΝΕΙ ΚΑΙ ΕΝΑ ΚΑΙ ΜΟΝΟ ΕΝΑ ΑΡΧΕΙΟ ΤΗΝ ΟΡΙΖΕΙ.
- Ένα αρχείο πηγαίου κώδικα δεν περιέχει extern δηλώσεις μεταβλητών. — τα αρχεία πηγαίου κώδικα πάντα περιλαμβάνουν το (μόνο) αρχείο κεφαλίδα που τα δηλώνει.
- Το αρχείο που ορίζει τη μεταβλητή την αρχικοποιεί επίσης. Η αρχικοποίηση με 0 είναι απαραίτητη.
- Αποφύγετε ολικές μεταβλητές όπου είναι εφικτό, χρησιμοποιήστε συναρτήσεις.