



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

Εισαγωγή στον δομημένο προγραμματισμό

Ενότητα 12^η: Δυναμική Ανάθεση Θέσης

Αν. καθηγητής Στεργίου Κώστας
e-mail: kstergiou@uowm.gr

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών



Πανεπιστήμιο Δυτικής Μακεδονίας



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Στόχοι της διάλεξης

- Περιγραφή της διαδικασίας δυναμικής δέσμευσης μνήμης και των σχετικών εντολών της C .



Περιεχόμενα

- Δυναμικές μεταβλητές.
- Δυναμική δέσμευση μνήμης.
 - Malloc.
- Αποδέσμευση μνήμης.
 - Free.
- Επανακαθορισμός δυναμικά δεσμευμένης μνήμης.
 - Realloc.



Στατική Δέσμευση Μνήμης

- Με τη δήλωση απλών μεταβλητών ο χώρος μνήμης που τους αποδίδεται παραμένει δεσμευμένος .
 - μέχρι το τέλος του προγράμματος .
 - ή μέχρι το τέλος του μπλοκ εντολών στο οποίο δηλώθηκαν.
- Η πρακτική αυτή ονομάζεται στατική δέσμευση μνήμης.
- Παρουσιάζει ιδιαίτερα προβλήματα εκμετάλλευσης της υπάρχουσας μνήμης καθώς δεν είναι πάντα εκ των προτέρων γνωστός ο όγκος της μνήμης τον οποίο θα κληθεί το κάθε πρόγραμμα να διαχειριστεί.



Παράδειγμα 1^ο

- Εκτιμούμε ότι ένα πρόγραμμα απαιτεί 100 πίνακες, διαστάσεων ο καθένας 10x10 το πολύ.
- Άρα χρειαζόμαστε αποθηκευτικό χώρο για 10.000 στοιχεία.
- Δηλώνουμε 100 διακριτούς πίνακες 10x10.
 - Κατά την διάρκεια της εκτέλεσης αποδεικνύεται ότι χρειαζόμαστε μόνο 10 πίνακες 11x10.
 - Το πρόγραμμά αποτυγχάνει παρότι επιθυμούμε να εκμεταλλευτούμε μόνο 1.100 στοιχεία από τα 10.000 που έχουμε ήδη δεσμεύσει.
- Αντιθέτως, εάν είχαμε δεσμεύσει χώρο για 10.000 στοιχεία σε μια «αποθηκευτική δομή», θα μπορούσαμε πολύ εύκολα να αναθέσουμε τους 10 πίνακες 11x10 .



Δυναμικές Μεταβλητές

- Οι μεταβλητές που δημιουργούνται κατά τη διάρκεια της εκτέλεσης ενός προγράμματος ονομάζονται δυναμικές μεταβλητές.
- Δημιουργούνται με κλήση στην συνάρτηση `malloc` για την ανάθεση συγκεκριμένου όγκου μνήμης.
 - η `malloc` βρίσκεται στο βασικό αρχείο `stdlib.h`
- Οι δυναμικές μεταβλητές ισχύουν μέχρι να δοθεί εντολή καταστροφής τους ή τον τερματισμό του προγράμματος.
- Οι δυναμικές μεταβλητές εκμεταλλεύονται πιο ευέλικτα τον διαθέσιμο χώρο μνήμης.



Δέσμευση Μνήμης

- <όνομα δείκτη μεταβλητής> = **malloc** (χώρος μνήμης);
- Η malloc επιστρέφει τη διεύθυνση της μνήμης που δέσμευσε ή NULL αν αποτύχει.
- Πριν την κλήση της **malloc** πρέπει να:
 - αποφασισθεί ο τύπος των δεδομένων τα οποία πρόκειται να αποθηκεύσουμε στη μεταβλητή.
 - δηλωθεί μια μεταβλητή δείκτη τύπου δεδομένων όμοιου με τον τύπο της δυναμικής μεταβλητής.
- Μετά την κλήση της **malloc**:
 - αποδίδεται στη μεταβλητή δείκτη η διεύθυνση της τοποθεσίας της μνήμης η οποία έχει δεσμευθεί για την αποθήκευση της δυναμικής μεταβλητής.
- Για την εύρεση του ζητούμενου χώρου μνήμης συνήθως καλείται η συνάρτηση **sizeof ()** .



Παράδειγμα 2^ο

```
/* Δήλωση ενός δείκτη σε χαρακτήρα */  
char *x;  
  
/* Δέσμευση χώρου μνήμης y χαρακτήρων */  
x = malloc(sizeof(char)*y);  
  
/* Στο δείκτη x αποθηκεύεται η διεύθυνση της  
μνήμης που δεσμεύτηκε */  
  
if (x == NULL) /* η δέσμευση απέτυχε */  
printf("Η δέσμευση απέτυχε\n");
```



Type Cast

- Παρότι αρκετοί compilers δεν το απαιτούν θα μπορούσε επίσης να υπάρχει η δήλωση μετατροπής (`char*`) για τον επιστρεφόμενο τύπο της `malloc`.
- `x = (char*) malloc (sizeof (char) * y) ;`
- η `malloc` επιστρέφει ένα δείκτη τύπου `void` ο οποίος για να αποδοθεί σε ένα δείκτη τύπου `char` πρέπει να μετατραπεί.



Αποδέσμευση Μνήμης

- Όταν η διαθέσιμη μνήμη είναι περιορισμένη ή οι δυναμικές αναθέσεις που ζητάμε είναι μεγάλων απαιτήσεων σε όγκο απαιτείται η ρητή ελευθέρωση των κομματιών της μνήμης που αποδόθηκαν δυναμικά και δεν είναι πλέον χρήσιμα στο πρόγραμμα.
- Για να αποδεσμεύσουμε τη μνήμη που δεσμεύτηκε με την κλήση της `malloc` χρησιμοποιούμε τη συνάρτηση `free`.
 - Βρίσκεται στο βασικό αρχείο `stdlib.h`.
 - `free (<όνομα δείκτη μεταβλητής>);`
- Μετά την κλήση της `free` συνίσταται η καταχώρηση της τιμής `NULL` στον δείκτη της μεταβλητής.
 - ο δείκτης συνεχίζει να υφίσταται και περιέχει μια διεύθυνση η οποία όμως δε μπορεί να χρησιμοποιηθεί.
 - `<όνομα δείκτη μεταβλητής> = NULL;`



Παράδειγμα 3^ο

```
/* Δέσμευση μνήμης y χαρακτήρων */
```

```
char *x;
```

```
x = malloc(sizeof(char) * y);
```

```
/* Αποδέσμευση μνήμης y χαρακτήρων */
```

```
free(x);
```

```
x = NULL;
```



Επανακαθορισμός Δέσμευσης Μνήμης

- Όταν έχουμε δεσμεύσει ένα χώρο μνήμης με κλήση της `malloc`, μπορούμε να επανακαθορίσουμε το μέγεθος της δεσμευμένης μνήμης με την συνάρτηση `realloc`.
 - Βρίσκεται στο βασικό αρχείο `<stdlib.h>`
 - `<νέος δείκτης> = realloc(<παλιός δείκτης>, <νέο μέγεθος>);`
- Αν ζητάμε αύξηση του χώρου μνήμης και υπάρχει δυνατότητα να γίνει στην παλιά θέση, επιστρέφεται η παλιά διεύθυνση πάλι, αλλιώς επιστρέφεται νέα διεύθυνση.
- Συνιστάται η αποθήκευση του νέου δείκτη σε νέα μεταβλητή ώστε να μην υπάρχει κίνδυνος να χαθεί ο παλιός χώρος μνήμης (αν η `realloc` επιστρέψει `NULL`).



Παράδειγμα 4^ο

```
/* Δέσμευση μνήμης γ χαρακτήρων */  
char *x, *x1;  
x = malloc(sizeof(char)*y);  
  
/* Επανακαθορισμός δεσμευμένης μνήμης */  
y1=y*2;  
x1 = realloc(x,sizeof(char)*y1);  
  
if (x1 != NULL)  
    x = x1;  
else  
    printf("Den uparxei arketi diathesimi  
mnimi\n");
```



Παράδειγμα: Δυναμική Δέσμευση Μνήμης για Μονοδιάστατο Πίνακα

```
/* Διάβασμα μεγέθους πίνακα */  
int megethos;  
int *pinakas;  
  
scanf ("%d", &megethos);  
  
/* Δέσμευση ανάλογου χώρου μνήμης */  
pinakas = malloc (megethos*sizeof (int));  
  
/* Απόδοση τιμών σε στοιχεία του πίνακα */  
pinakas [0]=5;  
scanf ("%d", &pinakas [1]);  
  
/* Υπενθύμιση: pinakas [i]==*(pinakas+i) */
```



Άσκηση 1^η

- Να γραφεί πρόγραμμα σε C που να ζητά από το χρήστη να καθορίσει ένα αριθμό n και να διαβάζει n πραγματικούς αριθμούς από την standard είσοδο. Στη συνέχεια, υπολογίζει το μέσο όρο των αριθμών αυτών και μετά για τον καθένα αριθμό το τετράγωνο της διαφοράς μεταξύ του αριθμού και του μέσου όρου (τυπική απόκλιση).
- Αν υποθέσουμε ότι το n είναι 3 και οι αριθμοί είναι 10, 5 και 3:
 - Ο μέσος όρος είναι $18/3 = 6$.
 - Τα τετράγωνα των διαφορών είναι:
 $(10-6)(10-6) = 16$, $(5-6)(5-6) = 1$, and $(3-6)(3-6) = 9$.
 - Έτσι, η τυπική απόκλιση είναι $(16+1+9)/3 = 26/3 = 8.6667$.



Άσκηση 2^η

```
#include <stdio.h> #include <stdlib.h>
main()
{   int n, i;
    double *values;
    double avg;
    double variance;
    printf("Set the amount of values:");
    scanf("%d", &n);
    if (n >= 0)
        values = (double *) malloc(sizeof(double)*n);
    for (i = 0; i < n; i++)
        scanf("%lf", &values[i]);
    avg = 0;
    for (i = 0; i < n; i++)
        avg += values[i];
    avg /= n;
    variance = 0;
    for (i = 0; i < n; i++)
        variance += ((values[i]-avg)*(values[i]-avg));
    variance /= n;
    printf("Average: %lf\n", avg);
    printf("Variance: %lf\n", variance);
    free (values);
}
```



Παράδειγμα: Δέσμευση Μνήμης Πίνακα Δομών

```
struct book {
    char *title;          /* Τίτλος */
    char *writer;        /* Συγγραφέας */
    int isbn;            /* Αριθμός ISBN */
    int code;           /* Κωδικός βιβλιοθήκης */
};

typedef struct book BOOK;
BOOK *books;
int amount;
printf("Δώσε πλήθος βιβλίων: ");
scanf("%d", &amount);
books = (BOOK *) malloc(amount*(sizeof(BOOK)));

if (books == NULL)
    printf("Η δέσμευση μνήμης απέτυχε\n");
```



Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Σημείωμα Αναφοράς

- Copyright Πανεπιστήμιο Δυτικής Μακεδονίας, Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών, Στεργίου Κωνσταντίνος. «Εισαγωγή στον Δομημένο Προγραμματισμό». Έκδοση: 1.0. Κοζάνη 2015. Διαθέσιμο από τη δικτυακή διεύθυνση: <https://eclass.uowm.gr/courses/ICTE258/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Όχι Παράγωγα Έργα Μη Εμπορική Χρήση 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Ως Μη Εμπορική ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό



Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

