Ανάπτυξη και Σχεδίαση Λογισμικού

Η γλώσσα προγραμματισμού C

Γεώργιος Δημητρίου

Δομές Δεδομένων και Κυρίως Πρόγραμμα

- Δομές δεδομένων εκτός από πίνακες:
  - Structs
  - Unions
  - Enums

- Κυρίως πρόγραμμα
  - Παράμετροι που περνιούνται από το λειτουργικό σύστημα
  - Έξοδος

Δομές Δεδομένων

- Structs
  - Για υλοποίηση βάσεων δεδομένων
- Unions
  - Για μείωση του χώρου αποθήκευσης
  - Για ηθελημένη διττή ερμηνεία του ίδιου αντικειμένου
- Enums
  - Για ακολουθίες σταθερών

Δομές Τύπου struct

- Οι βασικές δομές για δημιουργία εγγραφών βάσεων δεδομένων
- Μπορούν να έχουν όνομα (σε ειδικό χώρο ονομάτων) ή να είναι ανώνυμες
- Στην ουσία είναι ένθετες εμβέλειες στις μεταβλητές των οποίων η αναφορά γίνεται με συνδυασμό των ονομάτων της εμβέλειας και της μεταβλητής

Δηλώσεις Δομών struct

- Η λέξη-κλειδί struct, προαιρετικό όνομα και μια λίστα με δηλώσεις μεταβλητών σε άγκιστρα

```
struct product {
        int code;
        char name[30];
        float price;
};
```

προσοχή στο ';'

- Οι μεταβλητές ονομάζονται πεδία της δομής

Μεταβλητές Τύπου struct

- Η δήλωση μιας δομής δεν δημιουργεί μεταβλητές του τύπου της δομής
- Μεταβλητές ενός τύπου δομής δηλώνονται μετά τη δήλωση της δομής:

```
struct product p1,p2,p3;
```

  - Προσέξτε ότι η δήλωση απαιτεί και τη λέξη-κλειδί struct!
  - Η δήλωση της δομής και των μεταβλητών της μπορούν να συμπτυχθούν σε κοινή δήλωση

Αρχικοποίηση Δομών struct

- Όπως και στους πίνακες, απλά εδώ έχουμε διαφορετικούς τύπους στοιχείων

```
struct product p1 = { 3569, “milk”, 1.56 };

struct product p2 = { 3801, “lobster” };
```

Αναφορά σε Πεδία struct

- Με τη βοήθεια του τελεστή '.'
- Ένα πεδίο δομής struct λειτουργεί όπως όλες οι μεταβλητές

```
p2.price = 72.5;
printf("Price of product %d \"%s\"  = %f\n",
            p1.code, p1.name, p1.price);
scanf("%d", &p3.code);
```

Αναφορά σε Ολόκληρη Δομή

- Είναι δυνατή η χρήση ολόκληρης δομής ως ένα ενιαίο σύνολο τιμών
  - Σε αναθέσεις:

    p3 = p2;
  - Σε ορίσματα συναρτήσεων:

    void print_name(struct product);
  - Σε επιστροφές συναρτήσεων:

    struct product new_product(int, char[]);

Δείκτες σε Δομές struct

- Πρακτικοί για κλήσεις συναρτήσεων
  - Κυρίως όταν θέλουμε να τροποποιήσουμε κάποιο πεδίο της δομής παραμέτρου
    - Το προηγούμενο παράδειγμα δεν επιδέχεται τροποποίηση στη δομή όρισμα!

      void increase_price(struct product*, float);
  - Αλλά και στην επιστροφή συνάρτησης
    - Αντιγραφή δείκτη αντί ολόκληρης δομής!

      struct product * new_product(int, char[]);

Αναφορά σε Πεδία struct

- Όταν η μεταβλητή δομής είναι δείκτης, η αναφορά γίνεται με τον τελεστή '->'

  a->b αποτελεί σύντμηση του (*a).b

  και όχι του *a.b

```
void increase_price(struct product* p, float x)
{
    p->price += x;
}
```

Αν το p δεν ήταν δείκτης, η τροποποίηση θα παρέμενε τοπικά στη συνάρτηση!

Πίνακες Δομών struct

```
struct product products[100];

void inflation(float x) {
        int i;
        for (i=0; i<100; i++)
                products[i].price *= 1.0+x;
}
```

Ένθετες Δομές

- Ένα πεδίο δομής μπορεί να είναι τύπου δομής!

```
struct s1 { int d,m; };
struct s2 {
        char name[30];
        struct s1 birthday;
};
```

Αρχικοποίηση Ένθετων Δομών

- Με ή χωρίς ένθετα άγκιστρα

```
struct s2 friend = { “nikos”, {13, 7} };
struct s2 friend = { “nikos”, 13, 7 };
```

Πεδία Ένθετων Δομών

- Διαδοχικοί τελεστές '.' και '->' χωρίς περιορισμό:

```
friend.birthday.d = 20;
friend.birthday.m = 3;
```

Ανώνυμες Δομές

- Συνήθως σε ένθετες δομές τις οποίες δε χρησιμοποιούμε αλλού

```
struct s2 {
        char name[30];
        struct { int d,m; } birthday;
};
```

- Ή σε συνδυασμό με δήλωση typedef

Δήλωση typedef και Δομές

- Με δήλωση typedef το όνομα της δομής γίνεται περιττό, αφού δίνεται συνώνυμο σε όλο τον τύπο
  - και δε χρειάζεται έτσι δήλωση μεταβλητής με τη λέξη-κλειδί struct!

```
typedef struct { int d,m; } Bday;
Bday b_yours = { 20, 3 };
```

Δομές Τύπου union

- Ιδιαίτερες δομές της C, όμοιας σύνταξης με τις δομές struct, όπου όμως όλα τα πεδία μοιράζονται τον ίδιο χώρο στη μνήμη
  - Ίδια αρχική διεύθυνση αν πρόκειται για πεδία με τύπους διαφόρων μεγεθών
- Δίνοντας τιμή σε ένα πεδίο, δίνουμε τιμή σε όλα τα πεδία
  - Όμως αν τα υπόλοιπα πεδία είναι άλλου τύπου, η ανάθεση τιμής δεν έχει νόημα γι' αυτά!

Δηλώσεις Δομών union

- Όμοιες με τις δηλώσεις των δομών struct

```
union u {
        int akeraios;
        char xarakthres[4];
};

union u u1, u2;
```

Χρήση Δομών union

- Κατά λάθος ή επίτηδες;

```
u1.akeraios = -1;
printf("akeraios = %d\n", u1.akeraios);
```

-1

```
u1.xarakthres[0] = 'a';
printf("akeraios = %d\n", u1.akeraios);
```

-159 όχι πάντα…

Πώς μπορούμε να θυμόμαστε τι τύπο έχουμε αποθηκεύσει;

Χρήση Δομών union

- Συνήθως ένθετες δομές σε struct

```
struct s1 {
        int is_integer;
        union {
                int akeraios;
                char xarakthres[4];
        };
};
```

πεδίο ελέγχου της δομής union

ανώνυμη δομή union

Επιτυγχάνεται εξοικονόμηση χώρου!

Δομές Τύπου enum

- Ένας τρόπος να δώσουμε ονόματα σε ακολουθίες σταθερών
  - Για παράδειγμα μέρες της εβδομάδας

```
enum days { Monday, Tuesday, Wednesday,
            Thursday, Friday, Saturday, Sunday };

enum days d = Tuesday;
```

Δομές Τύπου enum

- Ο τύπος enum είναι ακέραιος τύπος, οι δε σταθερές που δηλώνονται θεωρούνται πως έχουν διαδοχικές τιμές από το 0, εκτός αν υπάρχει αρχικοποίηση

```
enum days { Monday = 1, Tuesday,
            Wednesday, Thursday, Friday = 10,
            Saturday, Sunday };
enum days d = Sunday;
```

← 12

Κυρίως Πρόγραμμα

- Προαιρετικές παράμετροι της συνάρτησης main()
  - int argc = Πλήθος ορισμάτων
  - char *argv[] = Πίνακας ονομάτων ορισμάτων
- Το πρώτο όνομα ορίζεται πάντα και είναι το όνομα του προγράμματος, άρα argc > 0
- Τα υπόλοιπα ονόματα είναι αλφαριθμητικά που η main() διαχειρίζεται με οποιονδήποτε τρόπο
  - Συνήθως είναι ονόματα αρχείων εισόδου/εξόδου ή συμβολισμοί επιλογών του χρήστη

Έξοδος Προγράμματος

- Κανονική έξοδος από το κυρίως πρόγραμμα μέσω της εντολής return
  - Αν δεν υπάρχει return στο τέλος, το πρόγραμμα θα τερματιστεί κανονικά σα να υπήρχε return
- Εναλλακτική έξοδος μέσω της συνάρτησης exit() της βιβλιοθήκης stdlib, που μπορεί να κληθεί από οποιοδήποτε σημείο του κώδικα
  - Η exit() έχει μία παράμετρο, έναν κωδικό εξόδου, συνήθως 0 για κανονική έξοδο

Τι Μάθαμε Σήμερα

- Τύπος struct
  - Δήλωση και αρχικοποίηση
  - Αναφορά στα πεδία μιας δομής
  - Δείκτες σε δομές και πίνακες δομών
  - Ένθετες δομές
- Τύπος union
- Τύπος enum
- Παράμετροι εισόδου/εξόδου της main()
- Συνάρτηση exit()