

Ανάπτυξη και Σχεδίαση Λογισμικού

Η γλώσσα
προγραμματισμού C

Γεώργιος Δημητρίου

Συναρτήσεις της C – Μέρος Β'

- Στοίβα και μηχανισμός κλήσης
- Αναδρομικές συναρτήσεις
- Στατικές μεταβλητές
- Άλλα θέματα

Μηχανισμός Στοίβας

- Δομή στοίβας στη μνήμη για να θυμόμαστε τη σειρά κλήσεων
 - Κάθε κλήση συνάρτησης δεσμεύει χώρο στη στοίβα για τις τοπικές μεταβλητές
 - Κάθε φωλιασμένη κλήση συνεχίζει με νέα δέσμευση
 - Κάθε έξοδος από συνάρτηση αφαιρεί από τη στοίβα τον πιο πρόσφατα δεσμευμένο χώρο
- Ο μηχανισμός στοίβας επιτρέπει πολλαπλές δεσμεύσεις για την ίδια συνάρτηση

Αναδρομικές Συναρτήσεις

- Μια συνάρτηση μπορεί να καλεί τον εαυτό της
 - Άμεσα ή έμμεσα
- Η κλήση προς συνάρτηση που ήδη εκτελείται δημιουργεί νέα δέσμευση στη στοίβα
 - Έτσι μπορούμε να έχουμε απεριόριστες αναδρομικές κλήσεις
- Η αναδρομή τερματίζεται με κάποια δομή ελέγχου

Παράδειγμα

```
int sumA (int n, int *a) {  
    if (n == 1) return *a;  
    return *a + sumA(n-1,a+1);  
}  
int main () {  
    int P[] = {10, 9, -4, 3, 0, -2, 8, 3, 1};  
    printf("sum = %d\n", sumA(9,P));  
}
```

Στατικές Μεταβλητές

- Τοπικές μεταβλητές που δεν αποδεσμεύονται με την έξοδο από τη συνάρτηση
 - Επομένως δε μπορούν να τοποθετούνται στη στοίβα
- Επιδέχονται αρχικοποίηση
 - Διαφορετικά αρχικοποιούνται σε μηδενική τιμή
- Με επόμενη κλήση της συνάρτησης βρίσκουμε τις προηγούμενες τιμές τους

Παράδειγμα

```
int func (int n, int *a) {
    static int firstTime = 1;
    int i;
    if (firstTime) {
        a = (int*)malloc(n*sizeof(int));
        for (i=0;i<n;i++) a[i]=i;
        firstTime = 0;
    }
    ... // υπόλοιπος κώδικας της func()
}
```

Πρωτότυπο Συνάρτησης (Δήλωση)

- Μια δήλωση χωρίς σώμα
 - Τερματίζεται με ';' μετά τη δήλωση παραμέτρων
 - Τα ονόματα των παραμέτρων είναι προαιρετικά
- Αποσκοπεί στην ενημέρωση του μεταγλωττιστή για την ύπαρξη της συνάρτησης πριν αυτή δηλωθεί πλήρως
 - Ο μεταγλωττιστής πρέπει να γνωρίζει τη συνάρτηση όταν αυτή καλείται!
- Έχει νόημα για έμμεση αναδρομή
 - Επιπλέον δίνεται καλύτερη εικόνα στον κώδικα

Παραδείγματα

```
int sumA (int, int*);
```

```
int func100 (int x, float *y);
```

```
void procB (double *[100], int);
```

Ημιτελείς Παράμετροι

- Παράμετροι τύπου πίνακα μπορούν να μη δηλώνουν το μέγεθος της πρώτης διάστασης
 - Στην ουσία περνούν ως δείκτες, οπότε το μέγεθος πρώτης διάστασης είναι αδιάφορο
 - Πώς όμως γνωρίζουμε το πλήθος στοιχείων;

```
void fun(int n, int a[][100][100]);
```

Καθολικές Μεταβλητές (Ξανά)

- Τοποθετούνται στη μνήμη εκτός στοίβας
 - Το ίδιο και οι στατικές μεταβλητές
- Λειτουργούν σαν τις στατικές, όμως τις βλέπουν όλες οι συναρτήσεις
 - Πλην επισκιάσεων
- Προγραμματιστικά είναι καλύτερα να χρησιμοποιούνται τοπικές μεταβλητές
 - Συνήθως δηλώνουμε μεγάλες δομές ως καθολικές για να μην γεμίζουμε τη στοίβα

Οι Παρενθέσεις ως Τελεστής

- Το ζεύγος παρενθέσεων ως τελεστής ‘()’ επιτρέπει να δούμε την κλήση συνάρτησης ως μέρος έκφρασης

```
x = x + a * *funY(y, &b);
```

- Προτεραιότητα ίδια με τους τελεστές ‘[]’, ‘.’, ‘->’
 - Η υψηλότερη από όλους τους τελεστές

ΔΕΙΚΤΕΣ ΣΥΝΑΡΤΗΣΕΩΝ

- Η συνάρτηση ως δομή δεδομένων
- Μπορούμε να περάσουμε μια συνάρτηση ως παράμετρο σε άλλη συνάρτηση

```
typedef int (*funp) (int,int,int[]);  
void F1 (funp f, int n, int a[]) {  
    ...  
    a[i] = x * (*f)(n,i-1,a);  
    ...  
}
```

Συναρτήσεις Βιβλιοθήκης

- Τα αρχεία βιβλιοθήκης της οδηγίας `#include` δεν περιέχουν τα σώματα των συναρτήσεων
 - Περιέχουν πρωτότυπα, ώστε να μπορεί ο μεταγλωττιστής να γνωρίζει τα ονόματα, τον τύπο και τις παραμέτρους των συναρτήσεων
 - Τα σώματα είναι ήδη μεταφρασμένα σε άλλα αρχεία και συνδέονται με το πρόγραμμα κατά τη φάση της σύνδεσης

Τι Μάθαμε Σήμερα

- Μηχανισμός κλήσης – Αναδρομή
- Στατικές μεταβλητές
- Πρωτότυπα συναρτήσεων
- Ημιτελείς παράμετροι
- Ο τελεστής ‘()’
- Δείκτες συναρτήσεων
- Συναρτήσεις βιβλιοθήκης