

Τρόποι περάσματος μεταβλητής σε κλήση απομακρυσμένης διαδικασίας.

Ήρθε η ώρα να κλείσουμε το κεφάλαιο "Ασκήσεις Εξεταστικής στα Καταναεμμεμένα", εεηγώντας Την τελευταία μας άσκηση.

Καταρχήν να πούμε τι εννοούμε όταν λέμε απομακρυσμένη διαδικασία.

Ας υποθέσουμε ότι έχουμε έναν εξυπηρετητή και έναν πελάτη.

Στον εξυπηρετητή είναι αποθηκευμένες κάποιες συναρτήσεις, μια από αυτές για παράδειγμα, η toSquare () η οποία ορίζεται ως εξής.

```
toSquare( int x)
{
  x= x*x;
  Return x;
}
```

η οποία δεν κάνει τίποτα άλλο από το να επιστρέφει την τιμή του τετραγώνου ενός αριθμού.

Έτσι μπορεί ο πελάτης να χρησιμοποιήσει αυτήν την συνάρτηση απλά καλώντας την.

Όταν γίνει αυτή η απομακρυσμένη κλήση, ο πελάτης στέλνει δύο πράγματα στον εξυπηρετητή. Την συνάρτηση που θέλει να χρησιμοποιήσει και την τιμή που θα έχει το όρισμά της.

Όταν λοιπόν στον πελάτη υπάρχει μια τιμή γ όπου γ=4, όταν ο εξυπηρετητής δεχτεί την κλήση από τον πελάτη θα πάει και θα ψάξει την αντίστοιχη συνάρτηση που είναι αποθηκευμένη μέσα του και θα την τρέξει βάζοντας σαν Όρισμα αυτήν την τιμή γ=4 ώστε να βρει το τετράγωνο του 4.

Υπάρχουν 3 τρόποι για να σταλεί (περαστεί) η τιμή από τον πελάτη στον εξυπηρετητή.

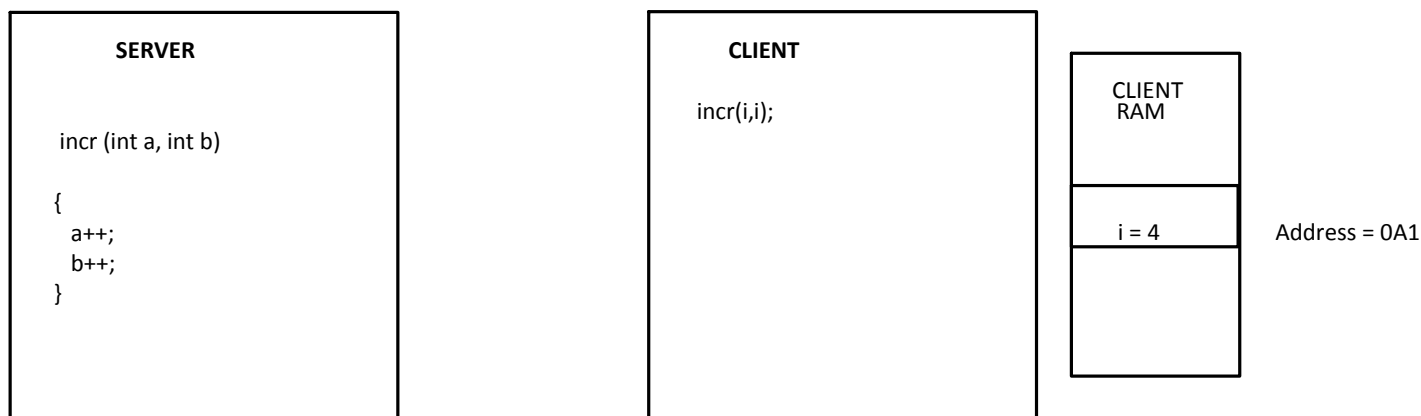
Ας δούμε όμως μια άσκηση που ζητείται αρκετά συχνά στις εξετάσεις, για να δούμε καλύτερα ποιοι είναι αυτοί οι τρόποι και πώς λειτουργούν.

#### Θέμα 4<sup>ο</sup> (25%)

Έστω ότι ένας server σε ένα καταναεμμεμένο σύστημα περιέχει τη ρουτίνα incr(a,b) η οποία σιξάνει την τιμή των a και b κακά 1 μέσα στον κώδικά της. Αν ο client καλέσει τη ρουτίνα αυτή με τη μέθοδο της απομακρυσμένης κλήσης (RPC) ως εξής incr(i,i), όπου η αρχική τιμή του i είναι 4, δώστε την τελική τιμή που παίρνει το i σε καθεμία από τις παραπτώσεις α) το πέρασμα των παραμέτρων γίνεται με τιμή και β) το πέρασμα των παραμέτρων γίνεται με αναγραφή - αποθήκευση. Δικαιολογήστε την απάντησή σας.

Στην παραπάνω εκφώνηση ζητούνται οι δύο τρόποι μόνο, Εμείς θα αναλύσουμε και τον τρίτο ..

Πάμε λοιπόν να φτιάξουμε το μοντέλο server -client που θα μας βοηθήσει να κατανοήσουμε καλύτερα τι γίνεται.



Αυτό είναι το μοντέλο που χρειαζόμαστε. Τώρα για χάριν ευκολίας θα γράφω μόνο τα περιεχόμενα του Server και ειδικότερα τον τρόπο που αλλάζει η τιμή των μεταβλητών μέσα στην συνάρτηση και στην μεριά του πελάτη θα γράφω μόνο την αντίστοιχη τιμή.

Για να ξεκινήσουμε λοιπόν.

Πέρασμα με τιμή. Σε αυτόν τον τρόπο, ο client στέλνει μόνο την τιμή της μεταβλητής στον server και όχι την ίδια την μεταβλητή (θέση μνήμης).

Έτσι έχουμε:

i=4

Incr(i,i) σαν να ήταν incr(4,4)

{  
i++; Για a παίρνουμε τώρα το πρώτο όρισμα i. Συνεπώς 4++ = 4+1 .. Άρα i=5

i=4

i++; Για b παίρνουμε τώρα το δεύτερο όρισμα i. Συνεπώς πάλι 4++ = 4+1 .. Άρα i=5  
}

i=4

Ποια είναι όμως η τελική τιμή του i, η τιμή δηλαδή που θα υπάρχει στην θέση μνήμης όπου βρίσκεται το i στον client; Όπως φαίνεται στο σχήμα, η τελική τιμή του i στον client παραμένει ίδια με την αρχική. Θυμηθείτε όπως είπαμε στην αρχή, ο πελάτης δεν στέλνει την ίδια την μεταβλητή αλλά μόνο την τιμή της. Έτσι ο server "δεν μπορεί να την βρει" και να την αλλάξει.

Πάμε να δούμε το πέρασμα με διεύθυνση. Δεν υπάρχει σε αυτήν την εκφώνηση αλλά πιθανότατα υπάρχει σε κάποια άλλη.

Σε αυτόν τον τρόπο, ο client στέλνει την ίδια την μεταβλητή στον server. Έτσι, κάθε φορά που αλλάζει η τιμή της μεταβλητής στον Server, γίνεται αυτόματα αλλαγή και στην θέση μνήμης του client που την περιέχει.

Έτσι έχουμε:

Incr(i,i) Εδώ το i δείχνει κατευθείαν την μεταβλητή i που βρίσκεται στον client.

Αρχικά η τιμή του i είναι 4. Άρα incr(4,4) κατά κάποιον τρόπο.

i=4

{  
i++; Για a παίρνουμε τώρα το πρώτο όρισμα i. Συνεπώς 4++ -> 4+1 .. Άρα i=5

i=5

i++; Για b παίρνουμε τώρα το δεύτερο όρισμα i. Θυμηθείτε ότι σαν ορίσματα έχουμε την ίδια την μεταβλητή και όχι την τιμή της. Επειδή όμως η τιμή του i άλλαξε αμέσως πριν και έγινε 5, για b θα πάρουμε την νέα τιμή της μεταβλητής. Συνεπώς 5++ -> 5+1 -> Άρα i = 6  
}

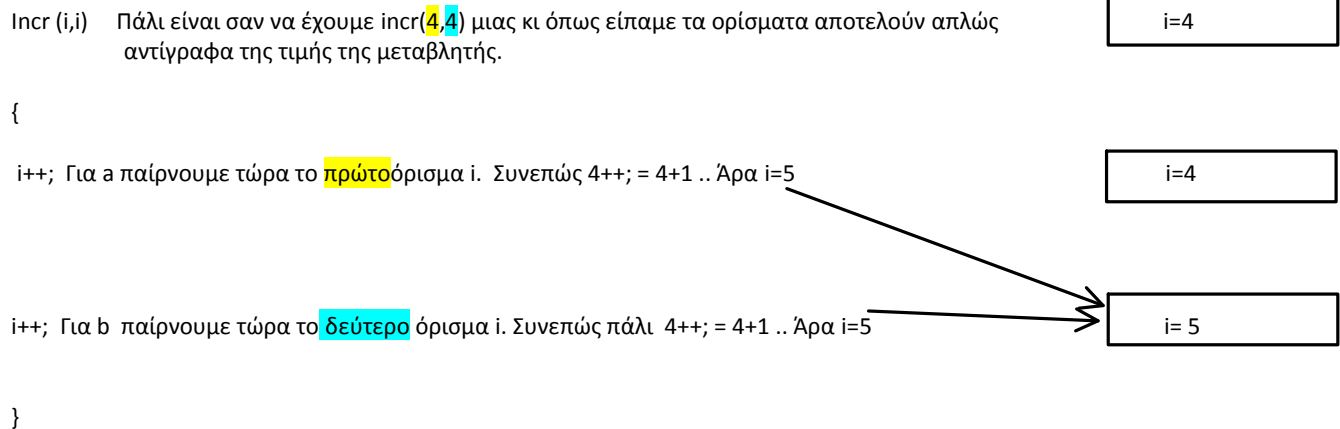
i=6

Σε αυτήν την περίπτωση δηλαδή βλέπουμε ότι η τελική τιμή γίνεται 6.

Τέλος στο πέρασμα με αντιγραφή και αποθήκευση, (ή αντιγραφή και επαναφορά), κατά την κλήση της συνάρτησης, αντιγράφεται σαν όρισμα ή τιμή της μεταβλητής ενώ παρέχεται και η πληροφορία της διεύθυνσης μνήμης που ανήκει αυτή. Όταν τελειώσει η συνάρτηση, μόνο η τελική τιμή που θα

Έχει αποκτήσει η μεταβλητή γυρίζει στον client και αποθηκεύεται.

Έτσι έχουμε:



Βλέπουμε ότι στο τέλος η τιμή του i γίνεται όσο και η τελική τιμή που παίρνει μέσα στην συνάρτηση.

Προσέξτε ότι όλες οι ενδιάμεσες αλλαγές του i δεν μεταφέρονται στην διεύθυνση μνήμης αλλά

Μόνο η τελική τιμή.

Έχουμε 2 βελάκια γιατί στην ουσία τα a και b είναι 2 ξεχωριστές μεταβλητές στην συνάρτηση και το ίδιο ισχύει και

Για τα αντίστοιχα i που έχουν πάρει την θέση τους. Εδώ οι δύο τελικές τιμές συμπίπτουν είτε επιστρέψει το πρώτο όρισμα πρώτα είτε το δεύτερο. Θα μπορούσαμε να είχαμε διαφορετική εντολή για το δεύτερο i για παράδειγμα, i=i+2;

Τότε η τιμή του δεύτερου i θα ήταν 6. Το ποια θα ήταν η τελική τιμή της μεταβλητής θα εξαρτιόταν από το ποιο από τα 2 ορίσματα i θα επέστρεφε πρώτα. Έτσι αν επέστρεφε τελευταίο το πρώτο όρισμα, αυτό που αντιστοιχεί στο a, η τελική τιμή του θα ήταν 5.

Στην αντίθετη περίπτωση θα ήταν 6. Είναι απαραίτητο λοιπόν να γνωρίζουμε με ποια σειρά επιστρέφουν οι τιμές στον πελάτη.

Αυτά είναι όλα όσα πρέπει να ξέρετε για τα περάσματα μεταβλητής σε RPC. Remote Process Call, στα ελληνικά, Κλήση απομακρυσμένης διαδικασίας.

Συνοπτικά, στο πέρασμα με τιμή, η πραγματική τιμή της μεταβλητής δεν αλλάζει καθόλου, στο πέρασμα με αναφορά, κάθε φορά που αλλάζει η τιμή στον server αλλάζει και η τιμή στον client, ενώ τέλος στο πέρασμα με αντιγραφή και αποθήκευση, όσο η συνάρτηση τρέχει, δεν επηρεάζεται η πραγματική τιμή αλλά μόλις αυτή τελειώσει, η τελική τιμή που θα έχει πάρει η μεταβλητή στην συνάρτηση, γυρίζει στον client και αποθηκεύεται ως τελική τιμή.

Και με αυτήν την άσκηση κλείσαμε το κεφάλαιο ασκήσεων.

Θα δούμε στην συνέχεια κάποια λίγα πράγματα για πιο θεωρητικά θέματα. Όχι τα πολύ βαριά, αλλά κάποια που είναι σχετικά εύκολα στην κατανόηση και πέφοτον και αρκετά συχνά.

Καλή Επιτυχία..