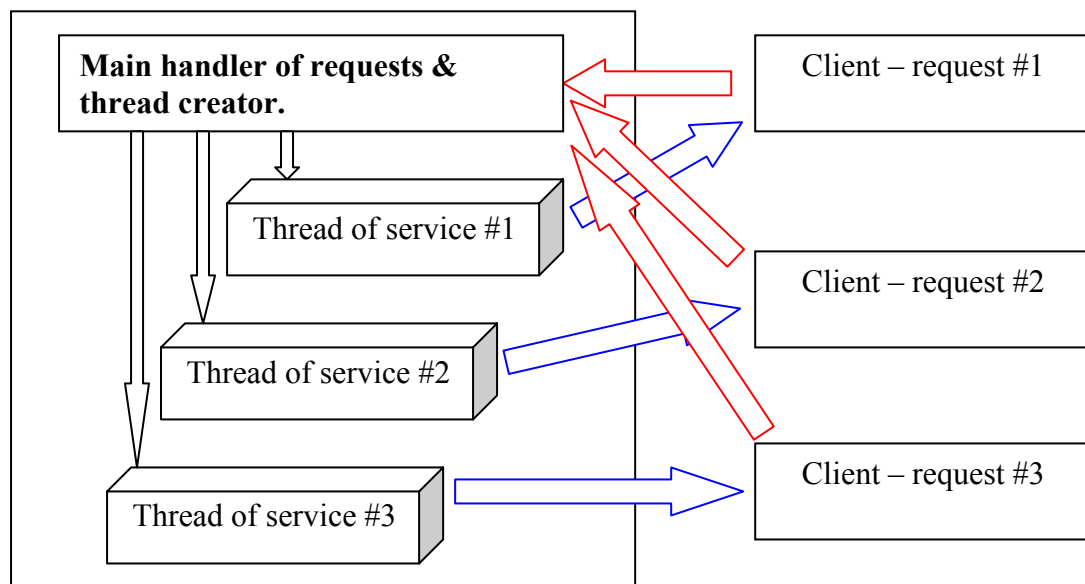


Ιωάννινα, 31 Μαρτίου 2003

### Κατανεμημένα Συστήματα Εργασία 3

**Άσκηση 3.9** Sketch the design of a multithreaded server that supports multiple protocols using sockets as its transport-level interface to the underlying operating system.

Server



**Σχήμα 1:** Λειτουργία Εξυπηρέτη

Δηλαδή στο παραπάνω σχήμα στον server υπάρχει ένας handler ο οποίος είναι υπεύθυνος για την επεξεργασία των αιτήσεων (κόκκινα βελάκια) από τους διάφορους πελάτες. Ο handler δεν επικοινωνεί άμεσα με τον πελάτη αλλά δημιουργεί ένα thread, το οποίο είναι κατάλληλο για την εξυπηρέτηση του πελάτη, και αυτό είναι εκείνο που θα επικοινωνήσει με τον πελάτη (μπλε βελάκια). Η επικοινωνία αυτή

μπορεί και είναι αμφίδρομη (απλά στο σχήμα το έχουμε κάνει μονόδρομο διότι το νήμα είναι αυτό που θα εκκινήσει την επικοινωνία τους, δηλαδή ο πελάτης θα κάνει “listen”).

Ειδικότερα εάν επιθυμούμε να κάνουμε χρήση των συναρτήσεων που χρησιμοποιούν τα sockets τότε τα βήματα θα είναι τα εξής:

- Ο πελάτης δεσμεύει ένα socket και κάνει μία κλήση προς τον server γνωστοποιώντας την ταυτότητά του, αλλά και την υπηρεσία που επιθυμεί να χρησιμοποιήσει.
- Στον εξυπηρέτη αναλαμβάνει ο handler να επεξεργαστεί την κλήση και δεν απαντά αμέσως, αλλά δημιουργεί ένα thread το οποίο είναι κατάλληλο για να εξυπηρετήσει τον πελάτη.
- Το thread απαντά στον πελάτη και έχουν στην διάθεση τους ένα socket για αμφίδρομη επικοινωνία. Ο handler πλέον δεν μπορεί να επικοινωνήσει με τον πελάτη εκτός και εάν έχει προβλεφθεί εκ των προτέρων κάτι τέτοιο.

### **Άσκηση 3.12 Mention some design issues for an object adapter that is used to support persistent objects.**

Μερικά σχεδιαστικά θέματα που θα πρέπει να κοιτάξει κανείς για την υλοποίηση ενός object adapter που χρησιμοποιείται για την υποστήριξη persistent αντικείμενα είναι:

- Η επιλογή του αλγορίθμου εξυπηρέτησης των αιτήσεων.
- Ο καθορισμός των δικαιωμάτων πρόσβασης των διαφόρων κόμβων σε κάθε αντικείμενο (π.χ. ποιο αντικείμενο μπορεί να κάνει Invoke ποιος κόμβος και ποιες λειτουργίες κτλ).
- Επιλογή χρήσης συγκεκριμένου port που θα ακούει ο εξυπηρέτης ή η επιλογή χρήσης δαίμονα που θα ακούει τις αιτήσεις που έχουν προορισμό τον εξυπηρέτη και στη συνέχεια θα τις προωθεί στον εξυπηρέτη.

- Ο καθορισμός του port που θα ακούει ο εξυπηρέτης ή ο καθορισμός του δαίμονα.
- Ο καθορισμός του χρονικού διαστήματος που ένα αντικείμενο θα παραμένει στη μνήμη μετά την κλήση του.
- Ο καθορισμός της μετακίνησης (ή κλωνοποίηση και μετακίνηση) ενός αντικειμένου ή όχι σε κάποιον άλλον κόμβο ύστερα από αίτηση κάποιου πελάτη ή του ίδιου του εξυπηρέτη καθώς επίσης και ο καθορισμός των συνθηκών κάτω από τις οποίες επιτυγχάνεται η μετακίνηση (ποιοι εξυπηρέτες έχουν δικαίωμα να ζητήσουν ποια αντικείμενα κτλ).
- Ο καθορισμός του τρόπου με τον οποίο θα γίνεται αναφορά στα αντικείμενα και του ονόματος με το οποίο θα αποθηκεύεται στον δίσκο.
- Ο καθορισμός της παράλληλης ή μη χρήσης των αντικειμένων από άλλους object adapter.
- Επιλογή πολιτικής που θα χρησιμοποιηθεί για το κλείδωμα των αντικειμένων (αφού ελεγχθεί αν απαιτείται κλείδωμα) στην περίπτωση της παράλληλης πρόσβασης πολλών εξυπηρετών.

**Άσκηση 3.17 Strong mobility in UNIX systems could be supported by allowing a process to fork a child on a remote machine. Explain how this would work.**

Ο τρόπος με τον οποίο σε συστήματα Unix υποστηρίζεται η strong mobility είναι επιτρέποντας μια διεργασία να κάνει fork και να δημιουργήσει ένα παιδί σε έναν απομακρυσμένο κόμβο (υπολογιστή). Ουσιαστικά δημιουργία με τη fork τοπικά ένα αντίτυπο της πατρικής διεργασίας και στην συνέχεια αντί να εκτελέσει κανονικές εντολές αναλαμβάνει να στείλει όλο τον κώδικά της αλλά και όλο το αντίστοιχο process address space στο remote μηχάνημα. Συνθήκη για να μπορέσει να λειτουργήσει κάτι τέτοιο είναι να πρόκειται για παρόμοιες μηχανές (να μπορούν να εκτελούν τον ίδιο κώδικα και να έχουν ίδια αναπαράσταση δεδομένων στην μνήμη).