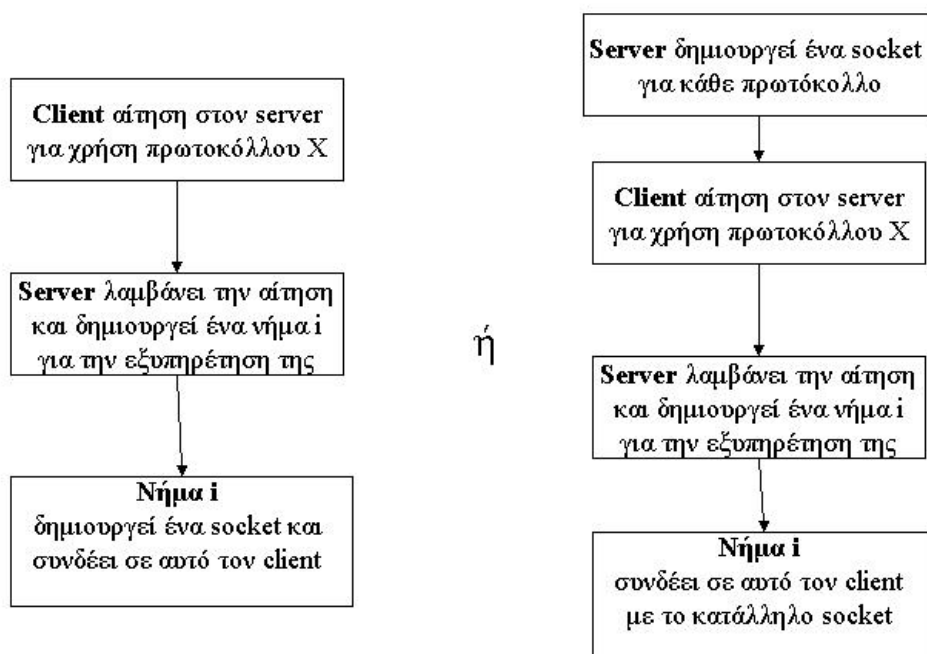


Άσκηση: Προβλήματα από το βιβλίο των Tanenbaum και Steen, κεφάλαιο 3: 9, 12 και 17

9. Σκιαγραφήστε τη σχεδίασή ενός πολυνηματικού εξυπηρετή ο οποίος υποστηρίζει πολλαπλά πρωτοκόλλα χρησιμοποιώντας sockets ως διεπαφή ανάμεσα στο επίπεδο μεταφοράς και το λειτουργικό σύστημα.

Στον server κάθε πρωτόκολλο δεσμεύει στατικά ή δυναμικά ένα socket. Υπάρχει πάντα ένα socket το οποίο ακούει τις αιτήσεις των πελατών για χρήση του τάδε πρωτοκόλλου και στη συνέχεια αναλαμβάνει να σηκώσει ένα Thread το οποίο είτε θα δώσει στον πελάτη την διεύθυνση του socket που ήδη έχει αντιστοιχισθεί στο πρωτόκολλο που θέλει να υλοποιήσει ο πελάτης είτε θα δημιουργήσει ένα socket για την υπηρεσία που του ζητείται και θα συνδέσει εκεί τον πελάτη. Η διαδικασία δηλαδή θα είναι αυτή που παρουσιάζεται στο παρακάτω σχήμα.



12. Σημειώστε κάποια θέματα σχεδίασης για έναν προσαρμογέα αντικειμένων (object adapter) ο οποίος χρησιμοποιείται για να υποστηρίξει «επίμονα» (persistent) αντικείμενα.

Ένα αντικείμενο ονομάζεται «επίμονο» αν η ύπαρξη του δεν συνδέεται με τον server ο οποίος το διαχειρίζεται. Με άλλα λόγια ένας εξηγητής (server) μπορεί να διαχειριστεί ένα τέτοιο αντικείμενο και στη συνέχεια το αποθηκεύει στη δευτερεύουσα μνήμη, όποτε αυτό συνεχίζει να υπάρχει και μετά τον τερματισμό του εξηγητή. Σχεδιάζοντας έναν προσαρμογέα αντικειμένων (object adapter) ο οποίος χρησιμοποιείται για να υποστηρίξει αυτού του είδους τα αντικείμενα θα πρέπει να σκεφτούμε τα εξής:

1. Τη στρατηγική ενεργοποίησης των αντικειμένων δηλαδή τον τρόπο με τον οποίο θα γίνει η κλήση στα αντικείμενα
2. Τον τύπο των μηνυμάτων που θα ανταλλάσσουν μεταξύ τους ο προσαρμογέας και οι πελάτες.
3. την δυνατότητα ή όχι παράλληλης διαχείρισης του ίδιου αντικειμένου από διαφορετικούς προσαρμογείς.
4. Τον τρόπο με τον οποίο θα πετύχουμε την απάλειψη συνθηκών ανταγωνισμού πάνω στα αντικείμενα τα οποία αφού χαρακτηρίζονται ως «επίμονα» υπάρχουν ανεξάρτητα με το αν κάποιος τα διαχειρίζεται και μπορεί ταυτόχρονα να θέλουν πολλοί να τα τροποποιήσουν.
5. Τον τρόπο με τον οποίο θα γίνεται η αναφορά στα αντικείμενα
6. Τον τρόπο με τον οποίο θα αποθηκεύουμε στη δευτερεύουσα μνήμη τα αντικείμενα μετά την χρήση τους.

17. Συστήματα UNIX μπορούν να υποστηρίξουν strong mobility αφήνοντας μια διεργασία να δημιουργήσει ένα παιδί με fork σε ένα απομακρυσμένο μηχάνημα. Εξηγήστε πως θα δούλευε αυτό.

Όταν το UNIX κάνει fork, δημιουργεί ένα αντίτυπο της διεργασίας που την καλεί και αντιγράφει εκεί τόσο τον κώδικα που εκτελεί η διεργασία όσο και την στοίβα εκτέλεσής της. Το UNIX επιτρέπει η γονική διεργασία να σταλθεί και να εκτελεστεί

σε ένα απομακρυσμένο μηχάνημα. Αν λοιπόν θέλουμε να μετακινηθεί μια διεργασία κάνουμε τα παρακάτω:

- την αναγκάζουμε να κάνει μια κλήση της `fork`
- να αναθέσει την εργασία που υπολείπεται στην θυγατρική διεργασία
- να στείλει τη θυγατρική να εκτελεστεί απομακρυσμένα

Το αποτέλεσμα θα είναι να συνεχιστεί στο απομακρυσμένο μηχάνημα η λειτουργία της διεργασίας και επίσης να έχω εκεί τόσο τον κώδικα (`code segment`) όσο και την στοίβα εκτέλεσης της (`execution segment`) αρά να υλοποιούμε `strong mobility`.