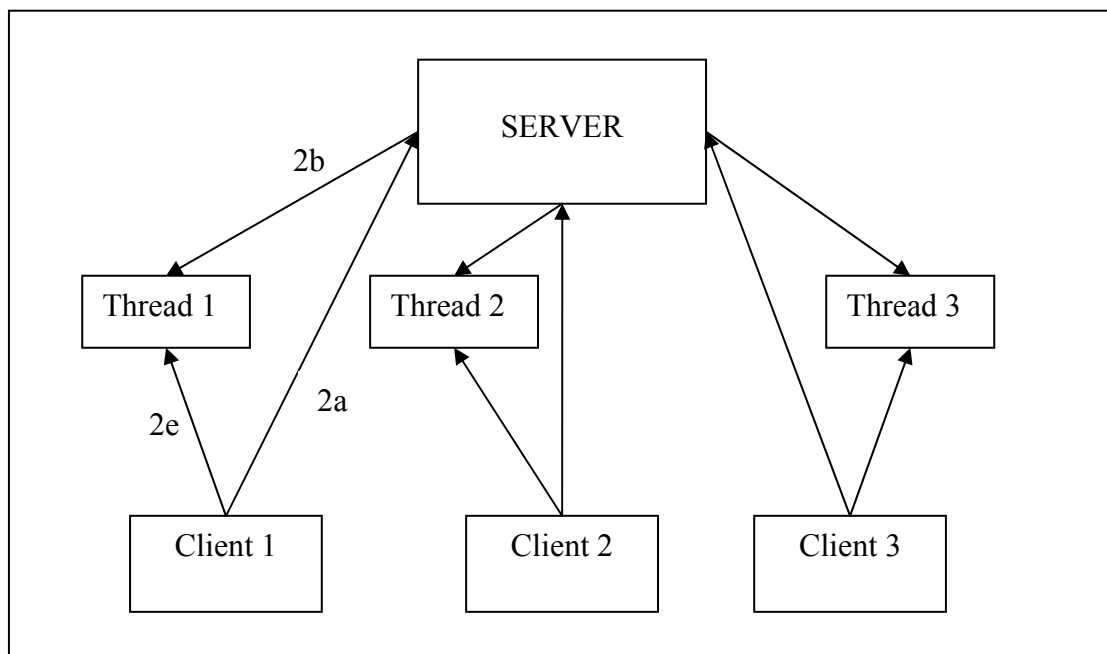


Άσκηση 9

Ένας τρόπος σχεδιασμού ενός τέτοιου server είναι η αντιστοίχιση ενός νήματος σε κάθε client. Η ιδέα δηλαδή είναι να δημιουργείται ένα νήμα από τον server για τη εξυπηρέτηση κάθε νέας αίτησης. Μετά την απάντηση της αίτησης το νήμα καταστρέφεται. Αναλυτικότερα τα βήματα είναι τα ακόλουθα (σχήμα 1):

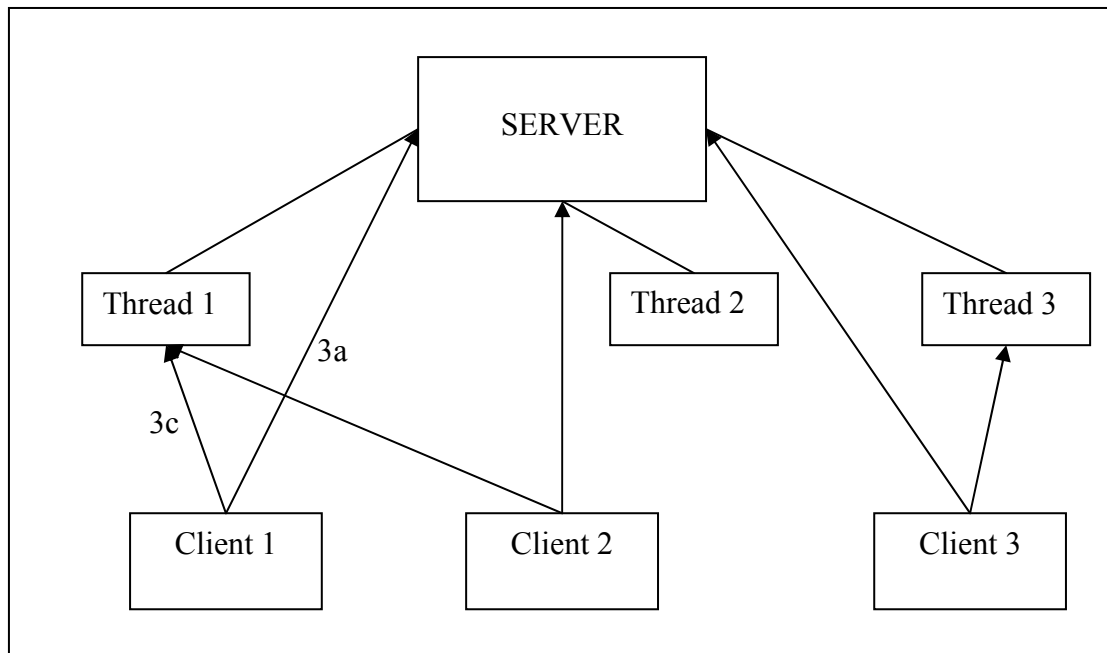
1. Αρχικά ο server με κλήση των συναρτήσεων socket και bind δεσμεύει ένα port στο οποίο θα δέχεται αιτήσεις από τους clients. Στην συνέχεια καλεί τις συναρτήσεις listen και accept και περιμένει να έρθουν αιτήσεις προς εξυπηρέτηση.
2. Για κάθε αίτηση που γίνεται από οποιονδήποτε client η διαδικασία είναι η εξής:
 - a. Ο client συνδέεται στο port στο οποίο δέχεται αιτήσεις ο server και ζητά επικοινωνία με κάποιο συγκεκριμένο πρωτόκολλο.
 - b. Ο server δημιουργεί ένα νήμα το οποίο μπορεί να επικοινωνήσει με χρήση του πρωτοκόλλου που ζητήθηκε από τον client.
 - c. Το νήμα με κλήση και πάλι των συναρτήσεων socket και bind δεσμεύει ένα port στο οποίο θα δεχτεί την αίτηση από τον client.
 - d. Ο server στέλνει πίσω στον client το port στο οποίο δέχεται αιτήσεις το νήμα που δημιούργησε.
 - e. Ο client συνδέεται στο νήμα και ξεκινάει την επικοινωνία με το πρωτόκολλο που χρησιμοποιεί.



Σχήμα 1

Ένας δεύτερος τρόπος σχεδιασμού είναι η αντιστοίχιση ενός νήματος σε κάθε πρωτόκολλο. Ο client συνδέεται στο νήμα του server το οποίο αντιστοιχεί στο πρωτόκολλο που ζητάει. Δηλαδή ο server από την αρχή της λειτουργίας του έχει δημιουργήσει έναν καθορισμένο αριθμό από νήματα (όσα και τα διαφορετικά πρωτόκολλα). Όλες οι αιτήσεις με το ίδιο πρωτόκολλο εξυπηρετούνται από το ίδιο νήμα (σειριακά). Αναλυτικότερα τα βήματα είναι τα ακόλουθα (σχήμα 2):

1. Αρχικά ο server δημιουργεί έναν αριθμό από νήματα (ένα για κάθε πρωτόκολλο). Επίσης δεσμεύει ένα port για τις εισερχόμενες αιτήσεις.
2. Κάθε νήμα δεσμεύει ένα port στο οποίο θα επικοινωνούν οι clients.
3. Για κάθε αίτηση που γίνεται από οποιονδήποτε client η διαδικασία είναι η εξής:
 - a. Ο client συνδέεται στο port στο οποίο δέχεται αιτήσεις ο server και ζητά επικοινωνία με κάποιο συγκεκριμένο πρωτόκολλο.
 - b. Ο server στέλνει πίσω στον client το port στο οποίο δέχεται αιτήσεις το νήμα που αναλαμβάνει το πρωτόκολλο που χρησιμοποιεί ο client.
 - c. Ο client συνδέεται στο νήμα και ξεκινάει την επικοινωνία με το πρωτόκολλο που χρησιμοποιεί.



Σχήμα 2

Άσκηση 12

Ένα αντικείμενο είναι *persistent* αν συνεχίζει να υπάρχει ακόμα και αν ο server σταματήσει να υποστηρίζει το service που προσφέρει το αντικείμενο αυτό. Δηλαδή ο server μπορεί να σώζει την κατάσταση ενός αντικειμένου σε ένα δευτερεύον μέσο αποθήκευσης, να τερματίζει την λειτουργία του και όταν επανεκκινήσει ο ίδιος ή κάποιος άλλος server, να μπορεί να διαβάσει την κατάσταση του αντικειμένου και να συνεχίσει την εξυπηρέτηση αιτήσεων που αφορούν αυτό το αντικείμενο.

Τα θέματα που προκύπτουν στην σχεδίαση του object adapter είναι:

- Την στρατηγική με την οποία θα σώζεται η κατάσταση στον δίσκο. Για παράδειγμα αν σώζονται κάθε 100 εγγραφές τότε υπάρχει περίπτωση ένα object να μην έχει ενημερωθεί για την αλλαγή που έχει κάνει ήδη ένα άλλο νωρίτερα σε μία κοινή resource. Επομένως ο object adapter θα πρέπει να είναι υπεύθυνος για την ενημέρωση όλων των object όσον αφορά τις αλλαγές που έχουν γίνει σε κοινά resources.
- Ο adapter πρέπει να φροντίζει ώστε να μην υπάρχουν προβλήματα ανταγωνισμού για την προσπέλαση κοινών resources από διαφορετικά objects. Δηλαδή αν δει ότι δυο αιτήσεις μπορεί να οδηγήσουν σε συνθήκη ανταγωνισμού με αποτέλεσμα πιθανή ασυνέπεια στα δεδομένα, θα πρέπει να σειριοποιήσει την εκτέλεση τους.
- Τι γίνεται όταν «πέσει» ο object adapter για κάποιο λόγο; Θα πρέπει με κάποιο τρόπο να κρατάει τα δεδομένα για τα objects που διαχειρίζεται. Σε αντίθετη περίπτωση δεν θα μπορεί να εξυπηρετήσει καμία αίτηση προς αυτά τα αντικείμενα όταν επανέλθει.
- Αν ο server σταματήσει να υποστηρίζει τα service όλων των objects του adapter τότε θα πρέπει ο adapter να συνεχίσει να υπάρχει;
- Θα πρέπει να είναι *persistent* και ο adapter; Δηλαδή να συνεχίζει να υπάρχει αν πέσει ο server στον οποίο αντιστοιχεί;

Άσκηση 17

Έστω A η διεργασία που θέλουμε να κλωνοποιήσουμε. Η διεργασία αυτή καλεί την *fork* και δημιουργεί μια διεργασία-παιδί B . Η A αντιγράφει την στοίβα της στην B . Για να εκτελεστεί η B σε έναν απομακρυσμένο υπολογιστή πρέπει να μεταφέρουμε τα εξής:

- Τον κώδικα ή ενδιάμεσο κώδικα (όπως συμβαίνει στην JAVA).
- Την στοίβα, ώστε η απομακρυσμένη διεργασία να έχει τις σωστές τιμές για τις μεταβλητές.
- Τον program counter, ώστε να συνεχιστεί η εκτέλεση από το σωστό σημείο.

Προσοχή πρέπει να δοθεί στο να εξασφαλίσουμε ότι τα δεδομένα που μεταφέρονται είναι συμβατά με την αρχιτεκτονική και το λειτουργικό σύστημα του απομακρυσμένου υπολογιστή. Θα πρέπει δηλαδή να γίνει τακτοποίηση (marshalling) των δεδομένων της διεργασίας ώστε να υπάρχει συμβατότητα στην εκτέλεση. Ένας τρόπος είναι να συμφωνηθεί ένα συγκεκριμένο πρωτόκολλο για την αναπαράσταση των παραπάνω δεδομένων.