ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

Παράδοση Ασκήσεων
Κεφάλαιο 3
Ασκήσεις ,9,12,17

Γεωργόπουλος Άλκης A.M.: 39
Κοντογιώργης Αναστάσιος A.M.: 43

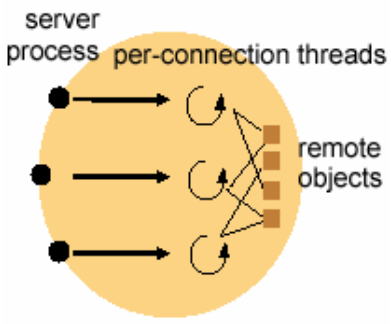
Ασκήσεις

9. Σκιαγραφήστε το σχεδιασμό ενός πολυνηματικού server ο οποίος υποστηρίζει πολλαπλά πρωτόκολλα χρησιμοποιώντας sockets για transport level interface με το λειτουργικό σύστημα.

Απάντηση

Ο ζητούμενος server θα εξυπηρετεί τις αιτήσεις των clients χρησιμοποιώντας νήματα. Η κύρια διεργασία του server θα δέχεται τις αιτήσεις για σύνδεση από τους clients και σε κάθε έναν από αυτούς θα αναθέτει ένα νήμα για να τους εξυπηρετήσει. Συγκεκριμένα ο σχεδιασμός του server στο transport layer θα είναι ως εξής:

1. Ο server ανοίγει ένα socket γνωστοποιώντας την IP του και δεσμεύοντας ένα port στο οποίο θα «ακούει» τις αιτήσεις από τους clients.
2. Κάθε φορά που συνδέεται κάποιος client προσδιορίζει το είδος της υπηρεσίας (πρωτόκολλο) που επιθυμεί από το server.
3. Σε κάθε εισερχόμενη σύνδεση δημιουργεί ένα thread και του αναθέτει να εξυπηρετήσει τις αιτήσεις του client. Εδώ υπάρχει ο σαφής περιορισμός, ότι ο μέγιστος αριθμός από threads που μπορεί να χρησιμοποιήσει ο server, είναι ο μέγιστος αριθμός συνδέσεων που ορίζεται από τη παράμετρο backlog στη κλήση listen του server.
   Δηλαδή η προτεινόμενη αρχιτεκτονική του πολυνηματικού server είναι “thread per client”, όπως φαίνεται στο παρακάτω σχήμα.



4. Το thread αυτό που αρχικοποιείται από το server αρχίζει τη λειτουργία του κάνοντας accept την αίτηση για σύνδεση του client.
5. Κάθε thread δουλεύει παράλληλα και ανεξάρτητα από τα υπόλοιπα και καταστρέφεται όταν ο client αποσυνδεθεί, όταν δηλαδή κλείσει το socket με το server.
6. Κάθε thread θα πρέπει να διαθέτει το δικό του χώρο μνήμης (stack – heap), ώστε να μην δημιουργούνται συνθήκες ανταγωνισμού με τα υπόλοιπα threads.

Επιπλέον θα πρέπει να γίνεται χρήση τεχνικών αμοιβαίου αποκλεισμού (π.χ. δυαδικοί σημαφόροι) κάθε φορά που προσπελαύνονται κοινοί πόροι.

12. Αναφέρατε κάποια θέματα σχεδιασμού για έναν adapter αντικειμένων ο οποίος χρησιμοποιείται για την υποστήριξη persistent αντικειμένων.

Απάντηση

Ουσιαστικά ο ζητούμενος adapter θα πρέπει να είναι επιφορτισμένος και με κάποιες επιπλέον ευθύνες ώστε να διατηρεί την persistency ιδιότητα των αντικειμένων.
Τα θέματα που προκύπτουν εδώ έχουν να κάνουν με την πληροφορία που θα πρέπει να κρατά ο adapter σχετικά με τη κατάσταση των objects.
Γενικά υπάρχουν δύο περιπτώσεις persistency, που σχετίζονται με το σχεδιασμό του adapter:

1. Persistency per object
   Σε αυτή τη περίπτωση κάθε φορά που θα τελειώνει μια εξυπηρέτηση για κάποιο object, ο adapter θα πρέπει να κρατάει πληροφορία για τη κατάσταση του object, ανάλογα με την υπηρεσία για την οποία χρησιμοποιείται.
   Αυτή η τεχνική εφαρμόζεται στο Globe service.
2. Persistency per client
   Σε αυτή τη περίπτωση ο adapter κρατάει πληροφορία, η οποία συσχετίζει μοναδικά τη κλήση του object η οποία εξυπηρετήθηκε με το client ο οποίος έκανε την αίτηση. Επιπλέον ο adapter θα πρέπει μπορεί να αναγνωρίσει κάποιο client ο οποίος έχει κάνει παλαιότερη αίτηση για συγκεκριμένο αντικείμενο, μέσω κάποιας persistence reference. Δηλαδή ο client κάθε φορά που κάνει μια αίτηση αποκτά κάποιο συνδετικό id με το αντικείμενο στο οποίο απευθύνθηκε.
   Αυτού του είδους οι αναφορές στο DCOM ονομάζονται monikers.

17. Η strong mobility στα συστήματα UNIX μπορεί να υποστηριχθεί επιτρέποντας μια διεργασία να κάνει fork ένα παιδί σε μια απομακρυσμένη μηχανή. Εξηγήστε πως μπορεί αυτό να γίνει.

Ας υποθέσουμε ότι μια διεργασία τρέχει στο κόμβο Α και θέλουμε να την κάνουμε fork στο κόμβο Β.
Το πρόβλημα εντοπίζεται στο τρόπο με τον οποίο μπορεί να εκτελεστεί ένα κομμάτι κώδικα της διεργασίας, που τρέχει στο Α, στο μηχάνημα Β.
Αυτό μπορεί να γίνει γενικά με δύο τρόπους.
Μία προσέγγιση είναι να σταλεί όλος ο κώδικας της γονικής διεργασίας στο Β και να εκτελεστεί από το σημείο που γίνεται fork και μετά. Αυτό προϋποθέτει ότι θα σταλεί επίσης και το state του προγράμματος που έτρεχε στο Α, όπως στοίβα, καθολικές μεταβλητές, heap κτλ. Αυτό θα πρέπει να προσαρμοσθεί στα δεδομένα της αρχιτεκτονικής του μηχανήματος Β, με χρήση κάποιου μηχανισμού marshalling.

Επιπλέον απαιτείται το μηχάνημα Β να διαθέτει compiler της C, ώστε να μεταγλωττίσει το κώδικα που του στάλθηκε και να δημιουργήσει το κατάλληλο χώρο στη μνήμη για να εκτελεστεί το πρόγραμμα.
Στην ειδική περίπτωση όπου Α και Β έχουν την ίδια αρχιτεκτονική, ο Α μπορεί να στείλει όλο το process state στο Β και αυτός να το εκτελέσει απευθείας από τον επεξεργαστή. Υπάρχει δηλαδή η δυνατότητα ο Α να στείλει τον εκτελέσιμο κώδικα και να τον στείλει μαζί με τη στοίβα του προγράμματος, τους καταχωρητές κτλ.
Αυτό ουσιαστικά μπορεί να γίνει ανεξαρτήτου αρχιτεκτονικής, δημιουργώντας μια εικονική μηχανή στο Β όπως θα γινόταν εάν το πρόγραμμα ήταν σε Java.