

Εργασία 2 – Ομαδική επικοινωνία

Αναπτύξτε λογισμικό ομαδικής επικοινωνίας που παρέχει λειτουργικότητα αξιόπιστης καθολικής πολυεκπομπής (reliable total group multicast). Το λογισμικό σας θα πρέπει να προσφέρει ένα κατάλληλο API, στο πνεύμα των παρακάτω λειτουργιών:

<code>int grp_setdir(char *ipaddr, int port);</code>	καθορισμός υπηρεσίας καταλόγου
<code>int grp_join(char *grpname, char *ipaddr, int port, char *myid);</code>	εγγραφή
<code>int grp_leave(int gsocket);</code>	διαγραφή
<code>int grp_send(int gsocket, void *msg, int mlen);</code>	αποστολή μηνύματος
<code>int grp_recv(int gsocket, int *mtype, void *msg, int *mlen);</code>	παραλαβή μηνύματος

Εσωτερικά, κάθε διεργασία διατηρεί ένα κανάλι επικοινωνίας με μια υπηρεσία καταλόγου (θα πρέπει να υλοποιηθεί και αυτή στο πλαίσιο της εργασίας) που χρησιμοποιείται για την διαχείριση της σύνθεσης της ομάδας, και άλλο ένα κανάλι επικοινωνίας για την αποστολή/παραλαβή μηνυμάτων στο πλαίσιο της ομάδας. Οι εσωτερικές διαδικασίες του λογισμικού που θα αναπτύξετε πρέπει να υλοποιούνται μέσα από ξεχωριστά νήματα εκτέλεσης, έτσι ώστε να υπάρχει απεμπλοκή από το νήμα της εφαρμογής (με κατάλληλο συγχρονισμό όπου αυτό χρειάζεται). Υποθέστε ότι οι διεργασίες δεν παρουσιάζουν βλάβες.

Στάδιο 1: group management

Όταν μια διεργασία επιθυμεί να προσχωρήσει σε μια ομάδα, καλεί την `join()` που στέλνει στην υπηρεσία καταλόγου τα στοιχεία της διεργασίας, και παραλαμβάνει τα στοιχεία όλων των υπολοίπων μελών της ομάδας. Η υπηρεσία καταλόγου ειδοποιεί τα υπόλοιπα μέλη της ομάδας για την προσθήκη, προτού στείλει επιβεβαίωση στο νέο μέλος. Τελικά η `join()` επιστρέφει έναν ακέραιο στο πνεύμα ενός socket file descriptor, μέσω του οποίου η εφαρμογή μπορεί να στείλει/λάβει μηνύματα. Αντίστοιχα, η `leave()` διαγράφει την διεργασία από την ομάδα, και η υπηρεσία καταλόγου ενημερώνει τα υπόλοιπα μέλη για την διαγραφή, προτού επιβεβαιώσει την διαγραφή στην απερχόμενη διεργασία. Η επικοινωνία με την υπηρεσία καταλόγου πρέπει να υλοποιηθεί σημείο-προς-σημείο με TCP/IP. Οι αλλαγές σύνθεσης πρέπει να ανακοινώνονται και στο επίπεδο της εφαρμογής, μέσω ειδικών μηνυμάτων που δημιουργεί αυτόματα το λογισμικό επικοινωνίας. Ο διαχωρισμός τους από τα μηνύματα που δημιουργεί η εφαρμογή πρέπει να γίνει μέσω του τύπου μηνύματος που επιστρέφει η `recv()`, ενώ η πληροφορία των μηνυμάτων αλλαγής σύνθεσης πρέπει να κωδικοποιείται με βάση κατάλληλη σύμβαση που να γνωρίζει η εφαρμογή.

Στάδιο 2: total multicast

Υλοποιήστε την αποστολή και παραλαβή μηνυμάτων με καθολική/κοινή σειρά μέσω `send()` και `recv()`. Η μετάδοση μηνυμάτων πρέπει να γίνεται με UDP/IP, και με εκμετάλλευση της δυνατότητας φυσικής πολυεκπομπής (βλέπε UDP multicast). Υποθέστε ότι η σύνθεση της ομάδας παραμένει σταθερή κατά την διάρκεια της διάδοσης των μηνυμάτων. Μετρήστε την απόδοση της υλοποίησής σας, μέσω μιας απλής εφαρμογής που στέλνει $K \gg 1$ συνεχόμενες φορές ένα μήνυμα με έναν μειούμενο σειριακό αριθμό, και για το τελευταίο μήνυμα (με σειριακό αριθμό 0) παραλαμβάνει μια επιβεβαίωση από κάθε άλλο μέλος. Καταγράψτε την συνολική καθυστέρηση ως συνάρτηση του K και του αριθμού N των μελών της ομάδας, π.χ., για $N=2,3,4$, για την περίπτωση που όλες οι διεργασίες είναι στο ίδιο μηχάνημα, και για την περίπτωση που κάθε διεργασία είναι σε ένα ξεχωριστό μηχάνημα.

Στάδιο 3: view synchronous message delivery (προαιρετικά)

Υποστηρίξτε αλλαγές σύνθεσης της ομάδας και κατά την διάρκεια της μετάδοσης μηνυμάτων εφαρμογής, στο πνεύμα του view-synchronous multicast. Σκεφτείτε τι αλλαγές πρέπει να γίνουν στην αλληλεπίδραση κάθε διεργασίας μέλους με τον κατάλογο καθώς των μελών μεταξύ τους, για να επιτευχθεί ο επιθυμητός συγχρονισμός.

Εφαρμογή επίδειξης

Δοκιμάστε και επιδείξτε το λογισμικό σας μέσω μιας εφαρμογής chat. Σκεφτείτε πως μπορείτε να εκβιάσετε την απώλεια μηνυμάτων ή/και καθυστερήσεις στην μετάδοση/επεξεργασία των μηνυμάτων, για να μπορείτε να κάνετε διάφορες δοκιμές.

Χρησιμοποιήστε όποια γλώσσα προγραμματισμού επιθυμείτε. Ο κώδικας σας πρέπει να μεταφράζεται και να εκτελείται κανονικά στο περιβάλλον Linux του εργαστηρίου. Εναλλακτικά, μπορείτε να επιδείξετε την υλοποίηση σε δικά σας laptop που θα πρέπει να συνδέονται στο ενσύρματο δίκτυο του εργαστηρίου. Σιγουρευτείτε ότι η επίδειξη σας δουλεύει σωστά.

Ημερομηνία παράδοσης: **Σάββατο 31 Μαρτίου 2018, 22:00.**

Παραδοτέα: (α) κώδικας με οδηγίες εγκατάστασης/χρήσης, (β) παρουσίαση του πρωτοκόλλου και της υλοποίησης, καθώς και των μετρήσεων και των συμπερασμάτων/παρατηρήσεων σας.