

Εργασία 2 – Ομαδική επικοινωνία με αξιόπιστη μετάδοση και καθολική σειρά παράδοσης μηνυμάτων

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

<code>int grp_join(char *grpname, char *myid);</code>	εγγραφή σε ομάδα
<code>int grp_leave(int g);</code>	διαγραφή από ομάδα
<code>int grp_send(int g, void *msg, int len, int catoc);</code>	αποστολή μηνύματος FIFO / CATOC
<code>int grp_recv(int g, void *msg, int *len, int block);</code>	παραλαβή μηνύματος blocking / non-blocking

Δεν επιτρέπεται να γίνουν υποθέσεις για το περιεχόμενο των μηνυμάτων της εφαρμογής. Επίσης, πρέπει να υπάρχει απεμπλοκή του νήματος της εφαρμογής από τους μηχανισμούς και τα νήματα του ενδιάμεσου λογισμικού έτσι ώστε αυτά να λειτουργούν ανεξάρτητα μεταξύ τους. Η εφαρμογή μπορεί να συμμετέχει σε πολλές ομάδες ταυτόχρονα.

Group management. Η διαχείριση της ομάδας γίνεται μέσω ξεχωριστής υπηρεσίας που εντοπίζεται μέσω UDP/IP multicast. Η λειτουργία `grp_join()` στέλνει στην υπηρεσία διαχείρισης τα στοιχεία της διεργασίας, και παραλαμβάνει τα στοιχεία των μελών που ήδη υπάρχουν στην ομάδα. Η υπηρεσία ειδοποιεί τα μέλη της ομάδας για την προσθήκη, προτού στείλει επιβεβαίωση στο νέο μέλος. Τελικά η `grp_join()` επιστρέφει ένα ακέραιο αναγνωριστικό (στο πνεύμα ενός file descriptor), μέσω του οποίου η εφαρμογή μπορεί στη συνέχεια να στέλνει/λαμβάνει ομαδικά μηνύματα. Η λειτουργία η `grp_leave()` διαγράφει την διεργασία από την ομάδα επικοινωνώντας με την υπηρεσία διαχείρισης, η οποία στην συνέχεια ενημερώνει τα υπόλοιπα μέλη για την διαγραφή. Η επικοινωνία ανάμεσα στα μέλη της ομάδας και την υπηρεσία διαχείρισης γίνεται με TCP/IP. Η υπηρεσία διαχείρισης δεν πρέπει να εμπλέκεται στην μετάδοση των μηνυμάτων ανάμεσα στα μέλη της ομάδας.

Group view. Σκεφτείτε πως η εφαρμογή θα ενημερώνεται από το λογισμικό της ομαδικής επικοινωνίας για την τρέχουσα σύνθεση και τις αλλαγές που γίνονται στην σύνθεση μιας ομάδας. Αλλάξτε / επεκτείνετε κατάλληλα την παραπάνω διεπαφή προγραμματισμού. Επίσης, σκεφτείτε τι χειρισμός απαιτείται σε επίπεδο του λογισμικού της ομαδικής επικοινωνίας κάθε φορά που γίνεται εγγραφή και διαγραφή ενός μέλους, έτσι ώστε να συνεχίζεται απρόσκοπτα η λειτουργία του εκάστοτε μηχανισμού μετάδοσης μηνυμάτων (βλέπε παρακάτω). Αποφασίστε αν και πως θα χειριστείτε το θέμα του view synchronicity.

FIFO reliable multicast. Τα μηνύματα που στέλνει η εφαρμογή μέσω της λειτουργίας `grp_send()` με παράμετρο `catoc=0` μεταδίδονται σε όλα τα μέλη της ομάδας αξιόπιστα στο πνεύμα του reliable multicast, ενώ παραδίδονται στην εφαρμογή μέσω `grp_recv()` με την σειρά αποστολής τους. Η μετάδοση γίνεται πάνω από UDP/IP, με κάθε μήνυμα να στέλνεται ξεχωριστά σε κάθε μέλος της ομάδας. Υποθέστε ότι κάθε μήνυμα χωράει σε ένα UDP datagram.

Causal total reliable multicast. Τα μηνύματα που στέλνει η εφαρμογή μέσω `grp_send()` με παράμετρο `catoc=1` μεταδίδονται όπως παραπάνω, αλλά παραδίδονται στην εφαρμογή με την ίδια σειρά σε όλα τα μέλη της ομάδας και έτσι ώστε αυτή η σειρά να είναι συμβατή με τη λογική σειρά δημιουργίας των μηνυμάτων. Χρησιμοποιήστε όποια μέθοδο/πρωτόκολλο επιθυμείτε.

Μετρήσεις. Μετρήστε την απόδοση της υλοποίησης σας ως συνάρτηση του αριθμού N των μελών της ομάδας ($N=2,3,4$), για την περίπτωση που μόνο (α) ένα μέλος και (β) όλα τα μέλη στέλνουν μηνύματα ταυτόχρονα. Καταγράψτε τον μέσο αριθμό μηνυμάτων που στέλνονται από το λογισμικό σας σε επίπεδο UDP/IP για κάθε μήνυμα που στέλνεται σε επίπεδο εφαρμογής. Καταγράψτε την καθυστέρηση μετάδοσης των μηνυμάτων εφαρμογής (σκεφτείτε και το πως θα το κάνετε αυτό).

Εφαρμογές επίδειξης. Επιδείξτε το λογισμικό σας μέσω μιας εφαρμογής chat και κατάλληλα σενάρια δοκιμών. Επίσης, σκεφτείτε και άλλους (πιο αυτοματοποιημένους) τρόπους να επιβεβαιώσετε την ορθότητα της υλοποίησης σας. Σκεφτείτε έναν απλό τρόπο να προσομοιώσετε ένα δίκτυο που χάνει ή καθυστερεί να μεταδώσει τα μηνύματα.

Βλάβες. Υποθέστε ότι οι διεργασίες δεν παρουσιάζουν βλάβες. Όμως, μια διεργασία μπορεί να καλέσει την `grp_leave()` και αποχωρήσει ανά πάσα στιγμή, διακόπτοντας τις όποιες παρασκηνιακές διαδικασίες αποστολής / παραλαβής μηνυμάτων.

Μπορείτε να χρησιμοποιήσετε όποια γλώσσα προγραμματισμού επιθυμείτε, αρκεί να έχει ρητή υποστήριξη για νήματα. Ακολουθήστε τις οδηγίες παράδοσης εργασιών (υπάρχουν στο eclass).

Φροντιστήριο/συζήτηση: **Πέμπτη 1 Απριλίου 2021**
 Ημερομηνία παράδοσης: **Σάββατο 17 Απριλίου 2021, 22:00**