



HY335: Δίκτυα Υπολογιστών
Χειμερινό Εξάμηνο 2012-2013
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης
Διδάσκουσα: Μαρία Παπαδοπούλη

Project 2012-2013

Υλοποίηση ενός chat server-client

Παράδοση: 7/2/2013

Περίληψη

Στο project αυτό θα υλοποιήσετε έναν chat client μαζί με τον αντίστοιχο server. Όλοι οι clients θα συνδέονται στον server, ο οποίος θα είναι υπεύθυνος για την προώθηση των μηνυμάτων στον κατάλληλο client. Ο κάθε client θα έχει ένα μοναδικό username, το οποίο το δηλώνει στον server όταν συνδεθεί για πρώτη φορά σε αυτόν. Έτσι οι υπόλοιποι clients θα στέλνουν σε αυτόν μηνύματα γνωρίζοντας μόνο το username του.

Τύποι Μηνυμάτων

Για να λειτουργήσει το σύστημά μας θα πρέπει να έχουμε ορίσει κάποια συγκεκριμένη μορφή στα μηνύματά μας. Κάθε μήνυμα θα έχει στην αρχή ένα byte με τον κωδικό του μηνύματος. Μετά τον κωδικό θα ακολουθούν τα κατάλληλα δεδομένα όπως ορίζει ο παρακάτω πίνακας

MSG TYPE (1B)	Description	MSG Body
0x01	Login message	Username
0x02	Logout message	
0x03	Text message	The text that was send
0x04	Username does not exist	An appropriate error message
0x05	User already logged in	An appropriate error message

0x06	Request logged in users	
0x07	Response with logged users	A message containing all the users currently logged in, in an appropriate form
0x08	Broadcast message	
0xFE	Unknown message type	
0xFF	General error	An error message describing why the operation failed

Είστε ελεύθεροι να προσθέσετε και τα δικά σας μηνύματα για οποιοδήποτε λόγο το κρίνετε απαραίτητο. Σε κάθε περίπτωση όμως **δεν θα πρέπει να υπάρχει η πληροφορία του μήκους του πακέτου μέσα στο μήνυμά σας**. Θα πρέπει να το παίρνετε προγραμματιστικά από το API της εκάστοτε socket βιβλιοθήκης.

ΠΡΟΣΟΧΗ: Για τα text messages και μόνο, μετά το message type θα πρέπει να ακολουθεί και το username του αποστολέα, καθώς και του παραλήπτη με την εξής μορφή:

3\$sender_username\$receiver_username\$body_of_the_text_msg

Υλοποίηση Server

Ο server ανάλογα με τα command line options που παίρνει θα πρέπει να κάνει τα εξής:

1. Θα πρέπει να δημιουργεί ένα **TCP** listening socket σε μια συγκεκριμένη port, στο οποίο θα συνδέονται όλοι οι clients που συνδέονται για πρώτη φορά. Την port θα την παίρνει μέσω του command line argument **-p**. Για παράδειγμα αν δωθεί στο prompt η εντολή,
>./chatserver -p 6886
η listening port του server θα πρέπει να είναι η 6886.
2. **>./chatserver -p 6886 -b**
Αν δωθεί το όρισμα **-b** θα πρέπει να υποστηρίζει automatic server discovery. Αυτό μπορεί να γίνει πολύ εύκολα απλά κάνοντας broadcast την IP του server αλλά και σε ποια port κάνει listen. Έτσι οι clients αυτόματα, θα μπορούν να συνδέονται, χωρίς να χρειάζεται ο χρήστης να δώσει την ip και την port του server. Βέβαια, υπάρχει περίπτωση άλλα προγράμματα να κάνουν το ίδιο, οπότε με κάποιον

τρόπο θα πρέπει να είστε σε θέση να ξεχωρίσετε τα δικά σας broadcast μηνύματα. Την port στην οποία θα κάνετε broadcast, θα την ορίσετε εσείς με κάποιο define στο header file, προσέχοντας όμως να μην ανήκει σε κάποιες από τις well known ports που χρησιμοποιούν διάφορα προγράμματα.

Πιο συγκεκριμένα ο server:

1. Θα ανοίγει ένα TCP listening socket το οποίο θα κάνει listen στην port που ορίζει το -p option, περιμένοντας για νέες συνδέσεις.
2. Αν έχει δωθεί το -b option, θα αρχίσει να κάνει broadcast την IP διεύθυνσή του και την port στην οποία κάνει listen, χρησιμοποιώντας το κατάλληλο format μηνύματος.
3. Για κάθε νέα incoming σύνδεση, θα δημιουργεί ένα νέο socket, το οποίο θα το διαχειρίζεται ένα νέο thread. Επίσης θα εκτυπώνει το εξής μήνυμα:
Client with IP xxxx just connected
όπου xxxx είναι η IP του client που συνδέθηκε.
4. Οποιοδήποτε μήνυμα και να έρθει ο server θα πρέπει να το αγνοεί αν ο συγκεκριμένος client δεν έχει κάνει πρώτα login. Επίσης θα πρέπει να εκτυπώνει στην οθόνη το εξής μήνυμα:
Ignoring message from client xxxx because is not logged in
όπου xxxx η IP του client.
5. Μόλις ο server λάβει ένα login message, ελέγχει αν ο συγκεκριμένος χρήστης είναι ήδη logged in. Αν όχι, τότε αποθηκεύει το username και το socket descriptor που του αντιστοιχεί. Αν είναι ήδη logged in ή υπάρχει κάποιος άλλος client με το ίδιο username, τότε επιστρέφει κατάλληλο μήνυμα σφάλματος.
6. Αν λάβει ένα request για τους logged in users, επιστρέφει κατάλληλο μήνυμα με τους online χρήστες.
7. Αν λάβει ένα text message ελέγχει αν ο παραλήπτης είναι online. Αν είναι, το προωθεί στον κατάλληλο παραλήπτη, διαφορετικά επιστρέφει κάποιο μήνυμα λάθους.
8. Αν λάβει κάποιο logout μήνυμα, απλά διαγράφει τον συγκεκριμένο χρήστη και τερματίζει την σύνδεση με αυτόν.
9. Επιπλέον ο server, κρατάει και τρεις counters. Ο πρώτος αποθηκεύει τον συνολικό αριθμό των μηνυμάτων που λήφθηκαν από τον server. Ο δεύτερος counter αποθηκεύει τον συνολικό αριθμό των bytes που λήφθηκαν, ενώ αντίθετα ο τρίτος κρατάει τον συνολικό αριθμό των bytes

που στάλθηκαν. Οι τιμές των counters θα πρέπει να τυπώνονται κάθε x δευτερόλεπτα (όπου το x το ορίζεται εσείς), από ένα ξεχωριστό thread.

Υλοποίηση client

Ο client ανάλογα με τα command line options που παίρνει θα πρέπει να κάνει τα εξής:

1. Θα πρέπει να συνδέεται σε έναν chatserver που ακούει σε μια συγκεκριμένη IP και port. Την IP, την port καθώς και το username θα πρέπει να τα παίρνει από κατάλληλα command line arguments. Για παράδειγμα στην εντολή, **`>./chatclient -a 192.168.1.10 -p 6886 -u mousakas`** η IP και η port του server είναι η 192.168.1.10 και 6886 αντίστοιχα, ενώ το username που θα χρησιμοποιήσει ο client είναι το mousakas.
2. Αν δεν δοθεί η IP και η port του server, θα πρέπει ο client να ξεκινήσει την διαδικασία του automatic server discovery, περιμένοντας κάποιο broadcast μήνυμα από τον server που περιέχει την πληροφορία που χρειάζεται για να συνδεθεί σε αυτόν.

Πιο συγκεκριμένα ο client θα πρέπει:

1. Να συνδέεται σε έναν chatserver στην IP και port που παίρνει ως παραμέτρους μέσω των ορισμάτων -a και -p αντίστοιχα, χρησιμοποιώντας TCP.
2. Αν δεν δοθούν τα -a και -p ορίσματα, τότε περιμένει μέχρι να λάβει κάποιο broadcast μήνυμα από τον server, που να περιέχει την IP και την port στην οποία πρέπει να συνδεθεί. Μόλις το λάβει συνδέεται κατευθείαν στον server.
3. Για να συνδεθεί στον server, ο χρήστης πρέπει να στείλει ένα login μήνυμα, πληκτρολογώντας απλά **login** στο prompt του client.
4. Για να πάρει ο χρήστης όλους τους συνδεδεμένους χρήστες, πληκτρολογεί στο prompt του client **users**.
5. Για να στείλει ο χρήστης κάποιο μήνυμα σε έναν άλλο user χρησιμοποιεί την εντολή sendto: ακολουθούμενη από το username του παραλήπτη ένα κενό και το κείμενο του μηνύματος.

6. Αντίστοιχα όταν ο client λαμβάνει ένα μήνυμα κειμένου από κάποιον άλλον χρήστη εμφανίζετε στην εξής μορφή:
from:sender_username message_body.
7. Αν δωθεί η εντολή **logout**, στέλνεται το κατάλληλο μήνυμα στον server, τερματίζεται η σύνδεση και ο client κλείνει.
8. Σε κάθε περίπτωση θα πρέπει να εκτυπώνονται κατάλληλα μηνύματα, για τα διάφορα μηνύματα λάθους που μπορεί να λαμβάνονται από τον server.

Οδηγίες

Το project θα πρέπει να υλοποιηθεί σε C ή C++ χρησιμοποιώντας τα posix sockets και όχι κάποια άλλη έτοιμη βιβλιοθήκη.

Σας δίνονται επίσης και τα header files καθώς και οι main συναρτήσεις. Είστε ελεύθεροι να προσθέσετε τις δικές σας συναρτήσεις/μεθόδους, αλλά θα πρέπει να υλοποιήσετε αυτές που υπάρχουν στα header files. **Δεν** μπορείτε επίσης να αλλάξετε τα ορίσματά τους, καθώς και τα command line arguments των main συναρτήσεων.

Το project σας θα πρέπει να είναι σε θέση να τρέχει σωστά σε Linux περιβάλλον. Είναι δική σας ευθύνη, αν προγραμματίζετε σε κάποιο άλλο λειτουργικό, να επιβεβαιώσετε πως ο κώδικάς σας τρέχει σωστά και σε Linux.

Μαζί με τον κώδικά σας, θα παραδώσετε οπωσδήποτε και ένα Makefile που θα κάνει build το project σας.

Για απορίες ή παρατηρήσεις χρησιμοποιήστε την λίστα του μαθήματος.