

Εργαστήριο Δικτύων Υπολογιστών
Δεύτερη άσκηση - Ακαδημαϊκό έτος 2013-2014

Ημερομηνία παράδοσης: 10/12/2013

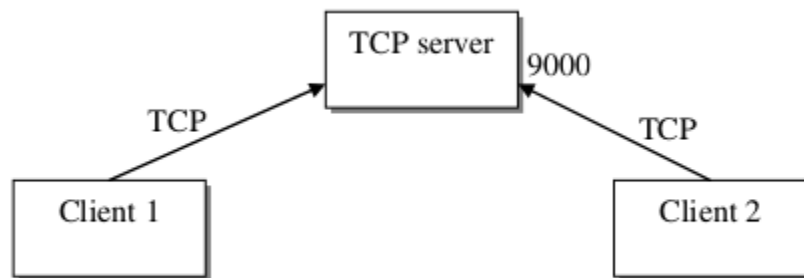
Παπαϊωάννου Κωνσταντίνος – 5052
Παυλόπουλος Φίλιππος – 5055

Αρ. Ομάδας: 24

Ανάλυση υλοποίησης πηγαίων αρχείων

- *Κώδικας του TCP Server (server.c)*

Αρχικά, φτιάχνουμε ένα TCP socket για την TCP επικοινωνία server-client και στη συνέχεια φτιάχνουμε τη struct (servaddr) με τα δεδομένα του server (IP, port) για να γίνει η σύνδεση με τον client. Στον server, για IP δίνεται το 0.0.0.0 ώστε να μπορεί να “ακούει” σε όλες τις IP. Δημιουργούμε τη σύνδεση με τον πρώτο client, με τη συνάρτηση accept() έχοντας εκτελέσει τη bind() και τη listen(). Με μια δεύτερη accept() γίνεται η σύνδεση με το δεύτερο client (σειριακά).

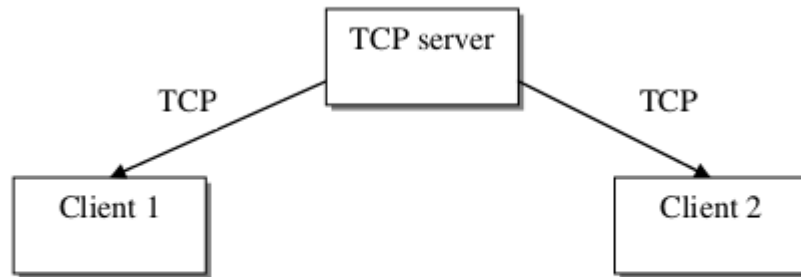


Εικόνα 2: Ο δεύτερος client συνδέεται στο server στο port 9000

Στη συνέχεια λαμβάνουμε, διαδοχικά, από τους δύο clients, τις structs με τα δεδομένα (IP και UDP port) που χρειάζονται για τη UDP επικοινωνία τους. Με την srand() παράγουμε τυχαία 0 ή 1, με τα οποία αποφασίζεται ποιος client θα γίνει UDP server και ποιος client και τους ενημερώνει (write()). Τέλος, στέλνει (write()) τις πληροφορίες του ενός client στον άλλο και εν συνεχεία κλείνει την σύνδεση με τους clients. Επιστρέφει στην πρώτη accept για να δεχτεί δυο νέους clients.

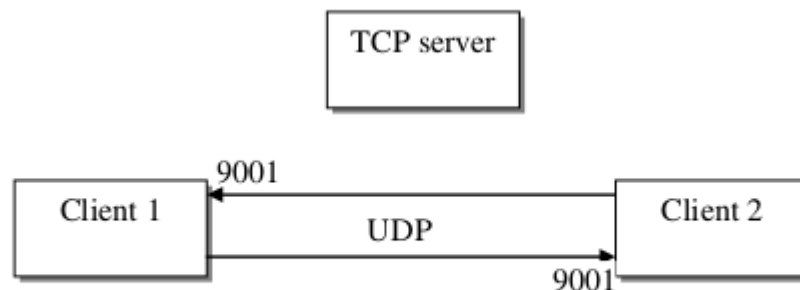
- *Κώδικας του TCP Client – των UDP Server/Client (client.c)*

Αρχικά, επαναλαμβάνουμε τη διαδικασία για την επικοινωνία TCP server-client (δημιουργία TCP socket) και στη συνέχεια φτιάχνουμε τη struct (servaddr) με τα δεδομένα του server (IP, port) για να γίνει η σύνδεση με αυτόν. Δημιουργούμε τη σύνδεση με τον TCP server με την connect(). Φτιάχνουμε μια struct (mine), με τις πληροφορίες του τρέχοντος client (IP, UDP port) και τη στέλνουμε στον server με τη write(). Ο πρώτος client που συνδέεται με τον server σε αυτό το σημείο μπλοκάρει μέχρι να συνδεθεί ένας δεύτερος. Μετά διαβάζουμε την επιλογή του server για το αν θα είναι UDP server ή UDP client και ανάλογα με την επιλογή, η διεργασία μπαίνει σε server ή client mode, έχοντας διαβάσει (με read()) τις πληροφορίες του άλλου TCP client που τρέχει παράλληλα. Οι πληροφορίες αυτές μπαίνουν στη struct other.



Εικόνα 3: Ο server καθορίζει (με τυχαίο τρόπο) ποιος από τους Client 1 και 2 θα έχει τον ρόλο του UDP Server και του UDP client και στέλνει την πληροφορία αυτή μαζί με την IP και τηνport (που ορίζεται κατά την έναρξη κάθε client) του UDP Server ώστε οι clients να επικοινωνήσουν μεταξύ τους, και εν συνεχεία κλείνει την σύνδεση με τους Clients.

Δημιουργείται ένα UDP socket για την επικοινωνία των δυο clients και ο UDP server περιμένει να δεχτεί (recvfrom()) μήνυμα από τον client, ενώ ο client στέλνει (sendto()), ασυγχρόνιστα και χωρίς σύνδεση (connectionless), μηνύματα στον server.



Εικόνα 4: Οι clients επικοινωνούν μέσω UDP. Αρχικά τυπώνεται ένα μήνυμα σε κάθε client για να ενημερωθούν οι χρήστες και στη συνέχεια αρχίζει η ανταλλαγή μηνυμάτων.

- *Γενικά*

Χρησιμοποιούμε έναν signal handler και στον server και στον client για να γίνεται ο τερματισμός (με CTRL-C) των προγραμμάτων, κλείνοντας σωστά τις συνδέσεις (close(sockfd)). ■