1st Assingment

Πρόλογος

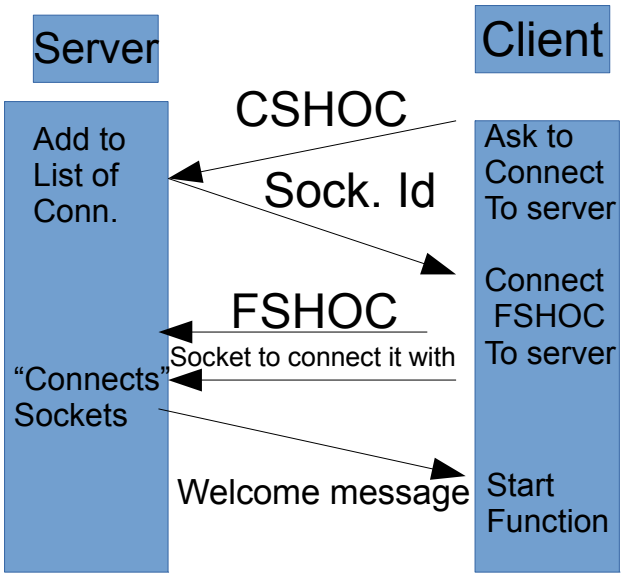
Παρακάτω παραθέτουμε πληροφορίες σχετικά με τη δημιουργία, την δομή και τη χρήση του προγράμματος που επισυνάπτουμε. Με λίγα λόγια το πρόγραμμα μας χρησιμοποιεί το TCP πρωτόκολλο επικοινωνίας και βασίζεται στη άμεση επικοινωνία client server ακόμα και για την επικοινωνία μεταξύ clients. Θα αναλύσω το τρόπο λειτουργίας του στη συνέχεια. Όποιος θέλει να εκτελέσει, παρ' όλα αυτά, το πρόγραμμα αρκεί να διαβάσει το αρχείο README.md όπου σύντομα αναφέρονται οι λειτουργίες του και η επιτρεπτή είσοδος από το χρήστη.

1.TCP έναντι UDP

Στο πρόγραμμα έγινε η χρήση του TCP πρωτόκολλου. Η επιλογή αυτή δικαιολογείται εν μέρει από την αξιοπιστία που παρέχει το TCP όσο αφορά την μεταφορά δεδομένων (εφόσων ενδιαφερόμαστε για τον διαμοιρασμό αρχείων μεταξύ των clients, το TCP αποτελεί τη καλύτερη λύση καθώς εγγυάται την ασφαλέστερη διεκπαιρέωση της εργασίας αυτής). Επίσης η διαρκής επικοινωνία με το server. που απαιτεί το TCP, δίνει στο πρόγραμμα μας πιο κατανοητή δομή και παρέχει μεγαλύτερο έλεγχο στις πληροφορίες που ανταλλάσουμε.

2.Γενική Δομή

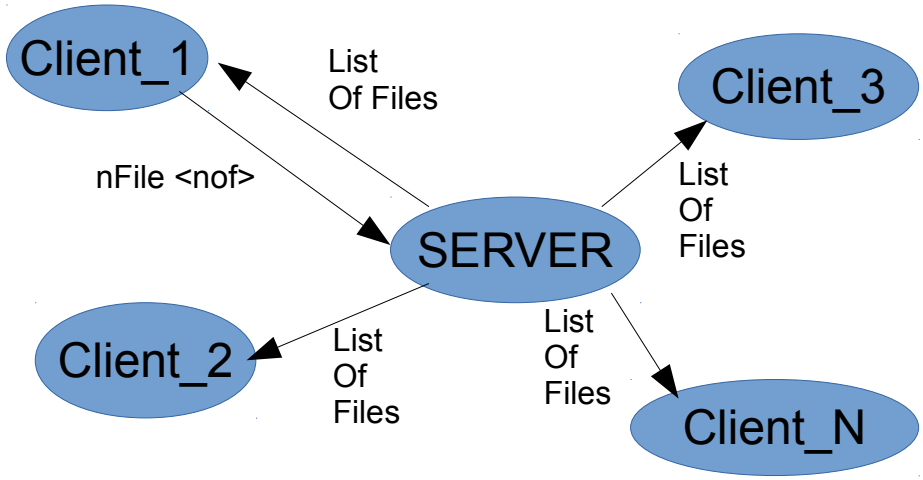
Κάθε client διαθέτει δύο sockets, την Chat-Socket(CSHOC) η οποία χρησιμοποιείται για την απλή επικοινωνία μεταξύ server-client και την FileExchange-Socket (FSHOC) η οποία είναι απαραίτητη κατά τον διαμοιρασμό των αρχείων. Κατά την σύνδεση του με τον client, ο server πρέπει να ενώσει τα δύο sockets(πχ. Σε ένα tuple) και να τα συσχετίσει με τον client. Πλέον ο μοναδικός τρόπος επικοινωνίας με τον client είναι αυτά τα δύο sockets τα οποία χρησιμοποιούνται ανάλογα με την ζητούμενη λειτουργία. Η συσχέτιση τους (CSHOC-FSHOC) και η σύνδεση client-server μπορεί να αναπαρασταθεί με το παρακάτω διάγραμμα:



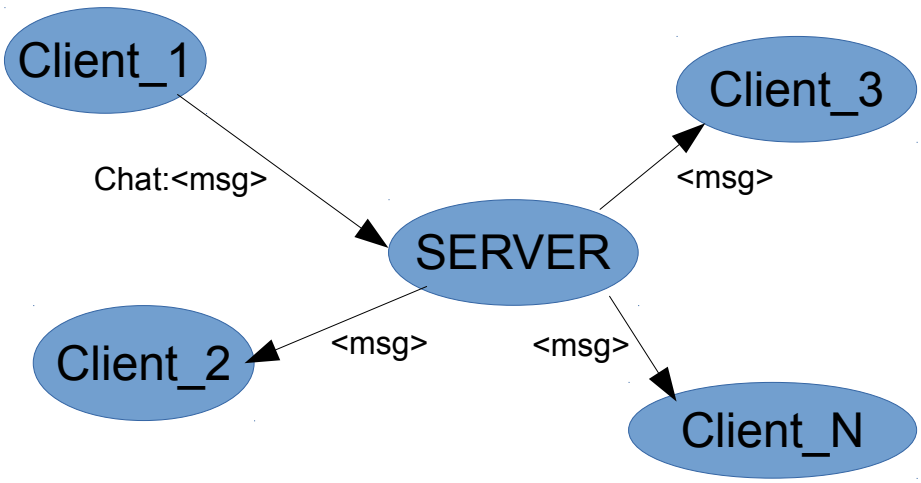
3. nFile/Chat:/ShowList/Q,q options (CSHOC)

Όλες αυτές οι λειτουργίες εκτελούνται μέσω των CSHOC των clients. Κάθε φορά που ένα τέτοιο αίτημα στέλνεται στο server κάνει διαβίβαση (pass-through) της απάντησης στους κατάλληλους παραλήπτες.

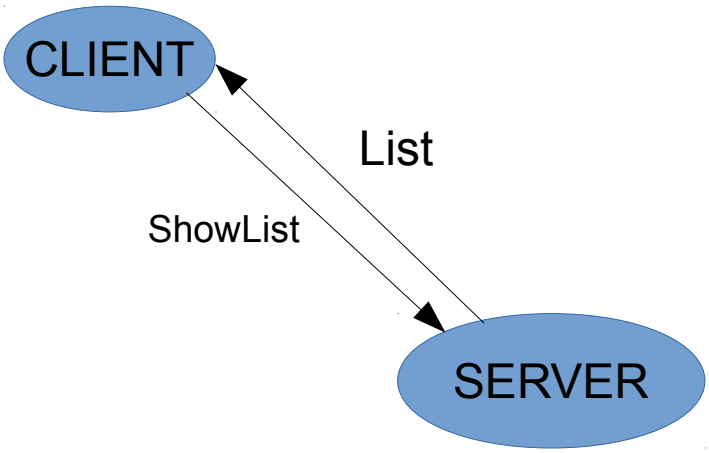
nFile <name of file> Ο client στέλνει το αρχείο που θέτει ελεύθερο προς διαμοιρασμό στον server. Ο server ανανεώνει τη λίστα των αρχείων του, συνδέει το συγκεκριμένο με τον client που το έστειλε, και στέλνει σε όλους τα αρχεία που είναι διαθέσιμα.



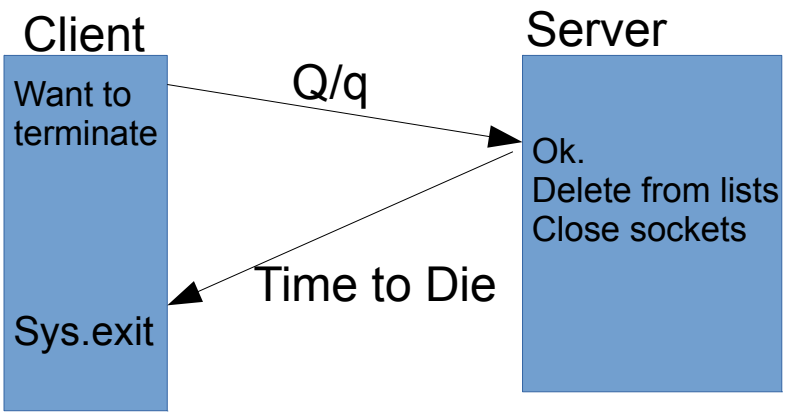
CHAT:<msg> Ο client στέλνει ένα μήνυμα που επιθυμεί να μοιραστεί με όλους τους διαθέσιμους clients, το μήνυμα αυτό στέλνεται στο server και ύστερα διαμοιράζεται σε όλους εκτός από τον ίδιο.



ShowList Ο client ζητά να δει τη λίστα με όλα τα διαθέσιμα αρχεία από τον server, ο server λαμβάνει την αίτηση του και ύστερα του επιστρέφει την εν λόγω λιστα.

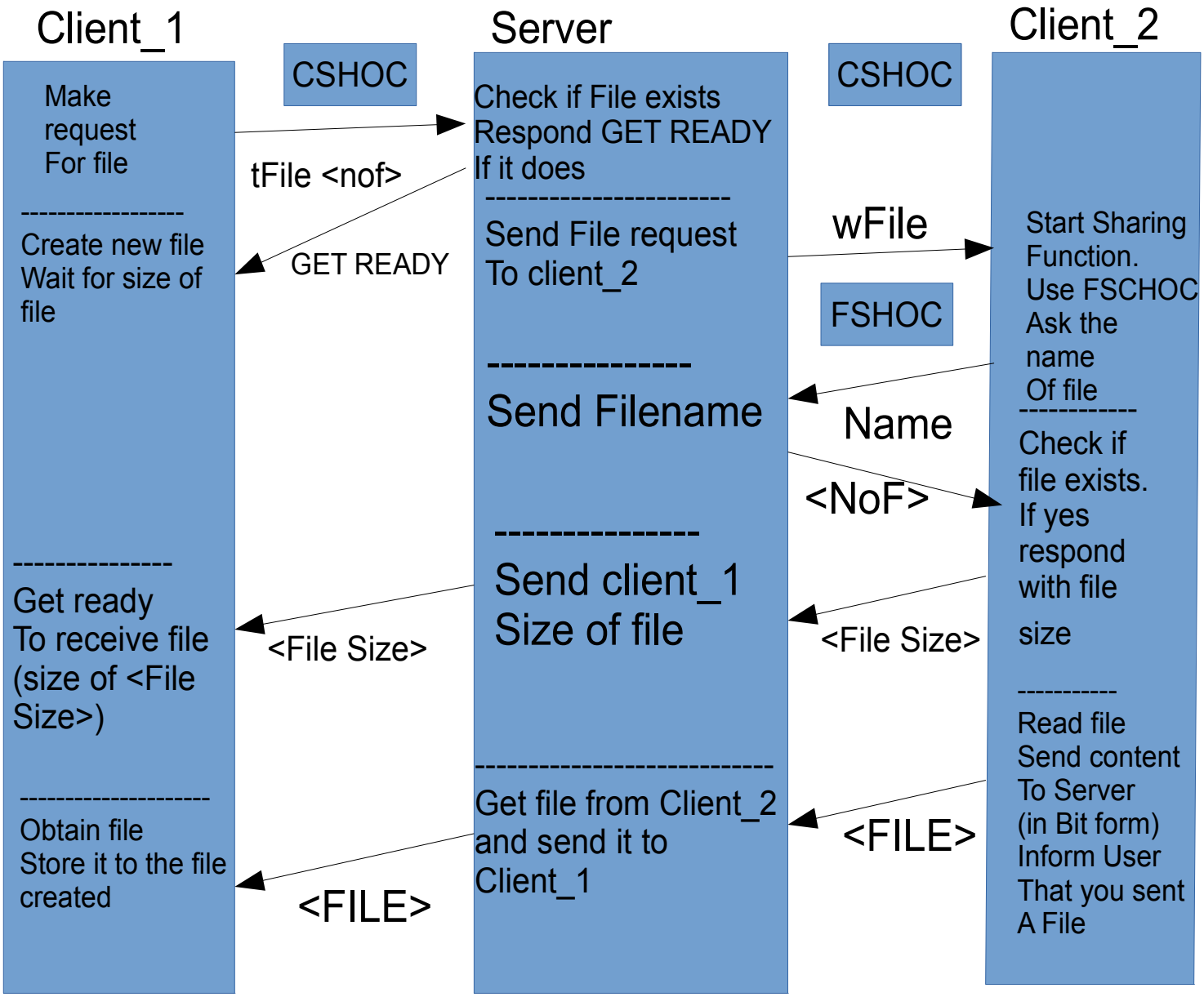


Q/q Ο client "ζητά" από τον server άδεια να τερματίσει. Ο server βλέποντας το αίτημα του client στέλνει το μήνυμα "Time to die", ώστε διαβάζοντας το ο client να τερματίσει, και έπειτα κλείνει τα socket του καθώς και "καθαρίζει" τα αρχεία που έχει ανεβάσει ο client από τη λίστα αποστολέας/αρχείο (listFilesConn). Ωστόσο στη λίστα με τα διαθέσιμα αρχεία (listFiles) παραμένουν αυτά που ανέβασε ο client πριν τερματίσει καθώς δεν δημιουργεί πρόβλημα οποιοδήποτε αίτημα για αυτά (απλά εμφανίζεται το μήνυμα "Sorry no such File uploaded. Maybe the client log out").



4. tFile option (FSHOC)

tFile <name of file> O client ζητά από το server κάποιο αρχείο που χρειάζεται (CSHOC) . Ο server παιρνει την αίτηση. Αρχικά ελέγχεται αν το αρχείο υπάρχει και στη συνέχεια επικοινωνεί με τον κάτοχο του αρχείου (CSHOC). Έπειτα ο client ρωτάει τον server για το όνομα του αρχείου (FSHOC) ο server αποκρίνεται και ο client του απαντάει με το μέγεθος (σε bytes) του αρχείου, αν όντως υπάρχει σε διαθεσιμότητα ή Not στην αντίθετη περίπτωση. Στις επόμενες ενέργειες ο server παίζει το ρόλο του διαμεσολαβητή όπου απλά μεταφέρει της πληροφορίες από τον ένα client (αποστολέα μέσω FSHOC), στον άλλον (παραλήπτης μέσω CSHOC).



5. BUGS etc.

Μερικές φορές, ενώ κάποιοι client είναι συνεδεμένοι στον server, μπορεί να αποσυνδεθούν μόνοι τους, με την έννοια ότι, δεν δέχονται και δεν μπορούν να στείλουν μηνύματα στο server. Παρά τις προσπάθειες μας να επιλύσουμε το πρόβλημα αυτό δεν βρήκαμε κάποια λύση (ή κάποιο στοιχείο για το τι φταίει). Ωστόσο το πρόγραμμα δείχνει να συμπεριφέρεται κανονικά(ο server εξυπηρετει τους άλλους clients) παρά αυτό το bug. Παρατηρήσαμε ότι είναι λιγότερο πιθανό να συμβεί αυτό, αν πρώτα συνδέσουμε όλους τους client και μετά στείλουμε οτιδήποτε στο server καθώς και αν κάθε αίτημα στο server σταλεί με μία αρκετά μεγάλη χρονική διαφορά (πχ. Τρία δευτερόλεπτα). Κατά τα άλλα το πρόγραμμα μας λειτουργεί κανονικά και εξασφαλίζει την επιτυχή μεταφορά αρχείων από τον ένα client στον άλλον.

6. Πηγές

Όσο αφορά τη βάση για τους κώδικες των client, server στηριχτήκαμε στα παραδείγματα στην ιστοσελίδα http://www.binarytides.com/python-socket-programming-tutorial/

Παραδείγματα προγραμματισμού σε python καθώς και γενική θεωρία, πάνω στη γλώσσα, στο παρακάτω link https://www.python.org/

Τέλος παράδειγμα για το διαμοιρασμό αρχείων μέσω server-client από τον ιστότοπο http://stackoverflow.com/questions/9382045/send-a-file-through-sockets-in-python