Κ24: Προγραμματισμός Συστήματος

Ονοματεπώνυμο: Κωνσταντίνος Πασχόπουλος

Αριθμός Μητρώου: 1115201500127

Ο κώδικας του server βρίσκεται στα αρχεία: dropbox_server.c, server_functions.c και server_functions.h. Ο κώδικας του client βρίσκεται στα αρχεία: dropbox_client.c, client_functions.c και client_functions.h. Στο αρχείο types.h βρίσκονται όλοι οι τύποι που έχω ορίσει και χρησιμοποιούνται από τον server και τον client.

Ο server διαχειρίζεται τις συνδέσεις με την χρήση της select μέσα σε ένα ατέρμων loop και καλεί την κατάλληλη συνάρτηση ανάλογα με το αίτημα που λαμβάνει. Όταν ο server δεχτεί αίτημα LOG_ON αποθηκεύει την IP με δυαδική μορφή στην λίστα, δηλαδή όχι σε numbers-and-dots notation. Αυτό το έκανα για να αποφύγω τις πολλές μετατροπές. Όταν δεχτεί αίτημα GET_CLIENTS ο server επιστρέφει τα στοιχεία όλων των clients, συμπεριλαμβανομένων και των στοιχείων του client που έστειλε το αίτημα. Στην συνέχεια ο client ελέγχει αν κάποιο από τα tuples που δέχτηκε είναι το δικό του και απλώς το αγνοεί.

Στον client έχω προσθέσει το όρισμα –m mirrorDir όπως στην $2^{η}$ εργασία, για να είναι πιο τακτοποιημένα τα αρχεία. Ο κάθε client αποθηκεύει στον φάκελο mirrorDir τα αρχεία που λαμβάνει από τους υπόλοιπους clients.

Για την δημιουργία του κοινόχρηστου κυκλικού buffer χρησιμοποίησα τις διαφάνειες του κύριου Ντούλα: http://cgi.di.uoa.gr/~antoulas/k24/lectures/l13.pdf. Ο client στέλνει ένα GET_CLIENTS αίτημα και βάζει όλους τους συνδεδεμένους clients που λαμβάνει σε μια κοινόχρηστη λίστα. Στην συνέχεια για κάθε client στην λίστα βάζει και ένα στοιχείο στον buffer με pathname = “-1” και version = “-1”, το οποίο υποδεικνύει στα threads ότι πρέπει να στείλουν αίτημα GET_FILE_LIST στον συγκεκριμένο client, αλλιώς πρέπει να στείλουν GET_FILE αίτημα.

Το κάθε thread εκτελεί την συνάρτηση worker. Όταν πρόκειται να στείλει ένα GET_FILE αίτημα το thread ελέγχει πρώτα αν το συγκεκριμένο αρχείο υπάρχει αποθηκευμένο στο mirrorDir. Αν δεν υπάρχει τότε θέτει το version ως -1, έτσι ώστε να καταλάβει ο άλλος client ότι πρέπει να στείλει το αρχείο. Αν υπάρχει αποθηκευμένο του στέλνει το version που έχει αποθηκευμένο. Το version το υλοποίησα με την Jenkins's one_at_a_time hash function: https://en.wikipedia.org/wiki/Jenkins_hash_function. Όποτε χρειάζομαι το version ενός αρχείου χρησιμοποιώ αυτή την συνάρτηση.

Όταν ένας client λάβει SIGINT στέλνει στον server LOG_OFF αίτημα και καθαρίζει την μνήμη. Για να το υλοποιήσω αυτό χρησιμοποίησα ένα flag του οποίου η τιμή αλλάζει μόνο μέσα στον signal handler. Έτσι το πρόγραμμα ελέγχοντας την τιμή του flag καταλαβαίνει ότι πρέπει να ξεκινήσει τον τερματισμό του. Στο LOG_OFF αίτημα ο client συμπεριλαμβάνει την IP και το port του για να μπορέσει να τον αναγνωρίσει σωστά ο server.

Τέλος για την διευκόλυνση της εξέτασης έχω συμπεριλάβει και τον φάκελο input με τον οποίο τέσταρα την εφαρμογή μου.

Για να γίνει compilation υπάρχει το Makefile το οποίο κάνει separate compilation.

Σημείωση: Για να τρέξει σωστά το πρόγραμμα μου πρέπει οι φάκελοι dirName και mirrorDir να δίνονται με relative και όχι absolute path. Ακόμη δεν πρέπει να υπάρχει '/' στο τέλος. Επίσης λόγω απροσεξίας μου σε όλα τα αιτήματα στέλνω πρώτα το port και μετά την IP. Για παράδειγμα αντί για: LOG_ON IP address, portNum στέλνω: LOG_ON portNum, IP address. Δεν αλλάζει σε κάτι την εργασία, απλώς ίσως προκαλέσει κάποια σύγχυση κατά την διάρκεια του ελέγχου.