

Δίκτυα Υπολογιστών

Προγραμματιστική Εργασία

Μάγκος Ραφαήλ-Γεώργιος 3100098

Μπογδάνος Μιχαήλ 3100123

Ντάτσεβ Ντάνιελ 3100131

Σημείωση: Ο κώδικας του παραδοτέου περιλαμβάνει και τα 2 τμήματα της εργασίας

I. Σύντομη περιγραφή της υλοποίησης

Η εφαρμογή του client ακολουθεί τα εξής βήματα:

- Συνδέεται με τον server
- Διαβάζει τον φάκελο dropbox (βρίσκεται πάντα στο current directory του προγράμματος) και συλλέγει το όνομα του αρχείου, το μεγεθός του και υπολογίζει το md5 hash value για το συγκεκριμένο αρχείο.
- Για κάθε αρχείο στο dropbox directory στέλνει ένα μήνυμα στον server της μορφής filename/filesize/md5sum/
- Μόλις ολοκληρωθεί η παραπάνω διαδικασία, ο client στέλνει έναν ειδικό χαρακτήρα στον server (End-Of-Transmission=0x04) που σηματοδοτεί το τέλος του μηνύματος που είχε να στείλει ο client (γενικότερα η χρήση του χαρακτήρα στην εφαρμογή του client ή του server σημαίνει το ίδιο πράγμα, το μήνυμα που ήθελε να στείλει ο αποστολέας ολοκληρώθηκε)
- Ο server θα στείλει την λίστα των αρχείων που επιθυμεί να λάβει από τον client (περισσότερες λεπτομέρειες στην περιγραφή του server) στη μορφή filename1/filename2/.../filenameN/. Ο client μόλις διαβάσει τον χαρακτήρα EOT, ξεκινά την επεξεργασία του μηνύματος η οποία περιλαμβάνει κατακερματισμό του μηνύματος με delimiter τον χαρακτήρα /. Κάθε token, που προέκυψε από την προηγούμενη διαδικασία, είναι ένα όνομα αρχείου. Γίνεται αίτημα στον server για κλείδωμα του αρχείου (ο client στέλνει το μήνυμα "rlock"). Αν το κλείδωμα πέτυχε ο server αποκρίνεται με το μήνυμα "ok" και ξεκινά η διαδικασία αποστολής. Η διαδικασία αποστολής περιλαμβάνει την κατά τμήματα ανάγνωση και αποστολή των δεδομένων στον server. Το μέγεθος του τμήματος καθορίζεται από το μέγεθος του buffer (ορίζεται στην αρχή της εφαρμογής). Πριν στείλει το επόμενο τμήμα, ο client ελέγχει αν ο server του γνωστοποίησε ότι έχασε το κλείδωμα στο αρχείο (lock timeout). Εάν έχει συμβεί κάτι τέτοιο, εμφανίζεται προειδοποίηση καθώς και η πιθανή αιτία που προκάλεσε αυτό το γεγονός (αργή σύνδεση ή/και μεγάλο αρχείο προς αποστολή) και ο συγχρονισμός αποτυγχάνει. Όμως, υπάρχει η περίπτωση το αρχείο να είναι ήδη κλειδωμένο από κάποιον άλλο client (ο server απάντησε με το μήνυμα "locked"). Αν συμβεί αυτό, η διεργασία του client θα μπλοκάρει στην κλήση της recv (χρησιμοποιούμε blocking sockets) και θα αφυπνιστεί όταν ο server στείλει το μήνυμα ξεκλειδώματος (μήνυμα "ok").

- Μετά από τα παραπάνω ο client θα περιμένει (με χρήση της sleep) ένα χρονικό διάστημα κάποιων δευτερολέπτων και θα ξανασυγχρονίσει τα αρχεία του με τον server.

Η εφαρμογή του server ακολουθεί τα εξής βήματα:

- Εκκινεί την σύνδεση και ακούει για νέες συνδέσεις στη θύρα 12010.
- Αρχικοποιεί ένα πίνακα νημάτων(σταθερού μεγέθους).Κάθε νήμα προορίζεται για την εξυπηρέτηση ενός client.
- Μόλις ένας client συνδεθεί,το main thread αναθέτει την εξυπηρέτησή του στο επόμενο νήμα που υπάρχει διαθέσιμο κοιτώντας τον πίνακα.
- Το νήμα ορίζει έναν buffer ώστε να διαβάσει τις πληροφορίες που έστειλε ο client για τα αρχεία που κατέχει ο τελευταίος.Το μήνυμα του client διαβάζεται ολόκληρο στον buffer.
- Ακολούθως,ο server κοιτάει κάθε κομμάτι του μηνύματος του client (κατακερματίζει το μήνυμα με βάση τον χαρακτήρα / όπως ακριβώς και ο client) και πρώτα ελέγχει αν υπάρχει αρχείο στο directory του με το όνομα που έστειλε ο client.Αν δεν υπάρχει,τότε σίγουρα πρέπει να συγχρονίσει αυτό το αρχείο οπότε και στέλνει το όνομα του αρχείου πίσω στον client.Αν υπάρχει το αρχείο,τότε πρέπει να υπολογιστεί το md5 hash value και να συγκριθεί με την τιμή που έστειλε ο client.Αν είναι ίδιες τότε δεν απαιτείται συγχρονισμός.Αν όμως διαφέρουν τότε ο server στέλνει το όνομα του αρχείου στον client.Έτσι δημιουργείται ένα μήνυμα από τα request του server (στη μορφή που περιγράψαμε στην ενότητα για τον client) που τερματίζεται από τον χαρακτήρα ΕΟΤ.Στα αρχεία που δεν απαιτείται συγχρονισμός,ο server «σβήνει» τις πληροφορίες των αρχείων αυτών στον buffer,βάζοντας χαρακτήρες NULL ώστε όταν ξαναδιαβάσει τον buffer να ξέρει ποια αρχεία περιμένει ότι θα λάβει από τον client.
- Έχοντας αποστείλει τα αρχεία που ζητάει από τον client,ο server διασχίζει τον buffer (αγνοώντας τους NULL χαρακτήρες για το λόγο που προαναφέραμε) και για κάθε αρχείο που περιμένει να λάβει,περιμένει μέχρι ο client να ζητήσει να το κλειδώσει.Εάν το κλείδωμα πετύχει,ξεκινάει ένα χρονόμετρο (ένα άλλο νήμα με παραμέτρους το νήμα που κάλεσε το χρονόμετρο,το socket και το file descriptor του αρχείου) το οποίο αν καταφέρει να λήξει τότε θα οδηγήσει σε τερματισμό το αρχικού νήματος,θα ξεκλειδώσει το αρχείο (lock timeout) και θα στείλει ένα σήμα ώστε να ξεκινήσει τη διαδικασία κλειδώματος του αρχείου κάποιο άλλο νήμα πελάτη που περιμένει(αν υπάρχει) .Το αρχικό νήμα του πελάτη ξεκινάει τη λήψη του αρχείου(γνωρίζοντας το μεγέθός του).Εάν καταφέρει να λάβει το αρχείο εντός χρόνου,τότε σταματάει το νήμα-χρονόμετρο και στέλνει ένα σήμα σε όλα τα νήματα ότι θα ξεκλειδώσει το αρχείο.Σε περίπτωση που το νήμα του πελάτη δεν καταφέρει να κλειδώσει το αρχείο,τότε γίνεται εισαγωγή του νήματος σε μια ουρά (FIFO),όπου εκεί θα περιμένει μέχρι να λάβει το σήμα ότι το αρχείο ξεκλειδώθηκε.

II. Περιγραφή αρχείων της εφαρμογής

DropboxClient.c : Το πηγαίο πρόγραμμα του πελάτη

FileSyncynchronizationClient.h : Αρχείο header μέσα στο οποίο δηλώνονται οι συναρτήσεις που χρησιμοποιεί ο πελάτης.

DropboxServer.c : Το πηγαίο πρόγραμμα του server

FileSynchronizationServer.h : Αρχείο header μέσα στο οποίο δηλώνονται οι συναρτήσεις που χρησιμοποιεί ο server.

Makefile : Και ο server αλλά και ο client έχουν από ένα makefile για πιο εύκολη μεταγλώττιση των προγραμμάτων. Η εφαρμογή γίνεται compile με την εντολή make clean σε κάθε ένα από τους φακέλους client και server. Το τρέξιμο του client γίνεται με την εντολή ./DropboxClient hostname (π.χ. ./DropboxClient localhost ή ./DropboxClient niovi.aueb.gr) Το τρέξιμο του server γίνεται με την εντολή ./DropboxServer
queue.h : Το αρχείο header που περιέχει την υλοποίηση της ουράς με χρήση συνδεδεμένης λίστας. Η υλοποίηση της ουράς βρέθηκε από τη σελίδα wikipedia.org

III. Αποκλίσεις από τις προδιαγραφές

- Η μέθοδος sync του client έχει το όνομα sync_ γιατί η πρώτη υπήρχε ήδη στην πρότυπη βιβλιοθήκη της C.
- Η μέθοδος listen του server έχει το όνομα serveClient γιατί η πρώτη υπήρχε ήδη στην πρότυπη βιβλιοθήκη της C για τα BSD sockets.
- Δεν υπάρχει αρχείο με όνομα SynchronizationConstants καθώς προσπαθήσαμε να είναι όσο το δυνατόν πιο ανεξάρτητες οι εφαρμογές, δηλαδή ο client να μπορεί να τρέχει από οποιοδήποτε υπολογιστή, χωρίς αναγκαστικά να υπάρχει ο server στον ίδιο φάκελο. Οι σταθερές που χρησιμοποιούνται από κάθε πρόγραμμα δηλώνονται στο ίδιο το πρόγραμμα.
- Δεν ήταν δυνατό να αποτρέψουμε οποιοδήποτε άλλο πρόγραμμα (για παράδειγμα έναν file editor) να επεξεργάζεται τα αρχεία που θέλει να συγχρονίσει ο client τη στιγμή που γίνεται ο συγχρονισμός. Αυτό συμβαίνει γιατί τα κλειδώματα στα λειτουργικά συστήματα που βασίζονται σε UNIX κλειδώνουν τα αρχεία με προαιρετικό τρόπο, δηλαδή πρέπει να συνεργάζονται και οι 2 εφαρμογές (και ο file editor και η δική μας εφαρμογή) ώστε να έχει νόημα το κλείδωμα και όχι να ανοίγουν πρώτα το αρχείο. Δηλαδή, αυτό που θα έπρεπε να γίνει είναι πρώτα να κοιτάνε αν υπάρχει κλείδωμα στο συγκεκριμένο αρχείο και αν δεν υπάρχει τότε να το ανοίγουν.

IV. Screenshots

1. Επιτυχημένη τροποποίηση και αποθήκευση του αρχείου στο server από κάποιον client.

Server:

```
Terminal - raphael@ubuntu: ~  
File Edit View Terminal Go Help  
niovi:~/comp.netw/server> make clean  
rm -f DropboxServer  
gcc -pthread -Wall -g DropboxServer.c -o DropboxServer -lssl -lcrypto  
niovi:~/comp.netw/server> ./DropboxServer  
Server socket created successfully.  
Bind operation was successful.  
Waiting for client to connect  
Connection accepted  
Waiting for client to connect  
Done reading files list  
Successfully received files list from client 1  
Client 1 requested lock.  
Trying to lock 1  
Receiving file test.txt 1  
Timer started!  
Unlocked file.  
Closed connection with client 1
```

Client:

```
Terminal - raphael@ubuntu: ~/Desktop/test/client  
File Edit View Terminal Go Help  
Sync with server has started.  
Checking file FileSynchronizationServer (another copy).h ...  
Checking file test (another copy).c ...  
Checking file test (copy).c ...  
Checking file FileSynchronizationClient.h ...  
Checking file DropboxServer.c ...  
Checking file test.txt ...  
Checking file DropboxServer (another copy).c ...  
Checking file FileSynchronizationServer.h ...  
Checking file DropboxServer (copy).c ...  
Checking file DropboxClient (another copy).c ...  
Checking file FileSynchronizationServer (copy).h ...  
Checking file FileSynchronizationClient (copy).h ...  
Checking file DropboxClient (copy).c ...  
Checking file test (another copy).txt ...  
Checking file DropboxClient.c ...  
Checking file FileSynchronizationClient (another copy).h ...  
Checking file test (copy).txt ...  
Checking file test.c ...  
Acquired lock for file test.txt.  
Done sending file test.txt.Total bytes sent 267
```

2. Λήξης του χρονομέτρου

Server:

```
niovi:~/comp.netw/server> ./DropboxServer  
Server socket created successfully.  
Bind operation was successful.  
Waiting for client to connect  
Connection accepted  
Waiting for client to connect  
Done reading files list  
Successfully received files list from client 1  
Client 1 requested lock.  
Trying to lock 1  
Receiving file random.txt 1  
Timer started!  
Timer done!Will cancel thread.  
Timer unlocked file.
```

Client:

```
Checking file test (copy).c ...
Checking file random.txt ...
Checking file FileSynchronizationClient.h ...
Checking file DropboxServer.c ...
Checking file test.txt ...
Checking file DropboxServer (another copy).c ...
Checking file FileSynchronizationServer.h ...
Checking file DropboxServer (copy).c ...
Checking file DropboxClient (another copy).c ...
Checking file FileSynchronizationServer (copy).h ...
Checking file FileSynchronizationClient (copy).h ...
Checking file DropboxClient (copy).c ...
Checking file test (another copy).txt ...
Checking file DropboxClient.c ...
Checking file FileSynchronizationClient (another copy).h ...
Checking file test (copy).txt ...
Checking file test.c ...
Acquired lock for file random.txt.
WARNING!Server will unlock file random.txt.Maybe the file is too big or the conn
ection too slow.Sync has stopped.
Client will exit.
raphael@ubuntu:~/Desktop/test/client$
```

3. Επιτυχές «κλείδωμα» του αρχείου (ίδιο με 1.)
4. Ανεπιτυχές «κλείδωμα» του αρχείου και τροποποίηση του από άλλον client

Server:

```
niovi:~/comp.netw/server> ./DropboxServer
Server socket created successfully.
Bind operation was successful.
Waiting for client to connect
Connection accepted
Waiting for client to connect
Done reading files list
Successfully received files list from client 1
Client 1 requested lock.
Trying to lock 1
Receiving file test.txt 1
Timer started!
Connection accepted
Waiting for client to connect
Done reading files list
Successfully received files list from client 2
Client 2 requested lock.
Trying to lock 2
Lock failed!2
```

Client1:

```
Sync with server has started.
Checking file FileSynchronizationServer (another copy).h ...
Checking file test (another copy).c ...
Checking file test (copy).c ...
Checking file FileSynchronizationClient.h ...
Checking file DropboxServer.c ...
Checking file test.txt ...
Checking file DropboxServer (another copy).c ...
Checking file FileSynchronizationServer.h ...
Checking file DropboxServer (copy).c ...
Checking file DropboxClient (another copy).c ...
Checking file FileSynchronizationServer (copy).h ...
Checking file FileSynchronizationClient (copy).h ...
Checking file DropboxClient (copy).c ...
Checking file test (another copy).txt ...
Checking file DropboxClient.c ...
Checking file FileSynchronizationClient (another copy).h ...
Checking file test (copy).txt ...
Checking file test.c ...
Acquired lock for file test.txt.
```

Client2:

```
Sync with server has started.
Checking file FileSynchronizationServer (another copy).h ...
Checking file test (another copy).c ...
Checking file test (copy).c ...
Checking file FileSynchronizationClient.h ...
Checking file DropboxServer.c ...
Checking file test.txt ...
Checking file DropboxServer (another copy).c ...
Checking file FileSynchronizationServer.h ...
Checking file DropboxServer (copy).c ...
Checking file DropboxClient (another copy).c ...
Checking file FileSynchronizationServer (copy).h ...
Checking file FileSynchronizationClient (copy).h ...
Checking file DropboxClient (copy).c ...
Checking file test (another copy).txt ...
Checking file DropboxClient.c ...
Checking file FileSynchronizationClient (another copy).h ...
Checking file test (copy).txt ...
Checking file test.c ...
File test.txt is locked.Waiting for unlock message..
```

Client1(τροποποιεί το αρχείο):

```
Sync with server has started.
Checking file FileSynchronizationServer (another copy).h ...
Checking file test (another copy).c ...
Checking file test (copy).c ...
Checking file FileSynchronizationClient.h ...
Checking file DropboxServer.c ...
Checking file test.txt ...
Checking file DropboxServer (another copy).c ...
Checking file FileSynchronizationServer.h ...
Checking file DropboxServer (copy).c ...
Checking file DropboxClient (another copy).c ...
Checking file FileSynchronizationServer (copy).h ...
Checking file FileSynchronizationClient (copy).h ...
Checking file DropboxClient (copy).c ...
Checking file test (another copy).txt ...
Checking file DropboxClient.c ...
Checking file FileSynchronizationClient (another copy).h ...
Checking file test (copy).txt ...
Checking file test.c ...
Acquired lock for file test.txt.
Done sending file test.txt.Total bytes sent 1890037
```

5. Επιτυχής αποδέσμευση του αρχείου και η χρήση του από άλλον client (συνέχεια του 4,ο client 2 λαμβάνει μήνυμα ότι το αρχείο ξεκλειδώθηκε και το επεξεργάζεται)

Client2:

```
Sync with server has started.
Checking file FileSynchronizationServer (another copy).h ...
Checking file test (another copy).c ...
Checking file test (copy).c ...
Checking file FileSynchronizationClient.h ...
Checking file DropboxServer.c ...
Checking file test.txt ...
Checking file DropboxServer (another copy).c ...
Checking file FileSynchronizationServer.h ...
Checking file DropboxServer (copy).c ...
Checking file DropboxClient (another copy).c ...
Checking file FileSynchronizationServer (copy).h ...
Checking file FileSynchronizationClient (copy).h ...
Checking file DropboxClient (copy).c ...
Checking file test (another copy).txt ...
Checking file DropboxClient.c ...
Checking file FileSynchronizationClient (another copy).h ...
Checking file test (copy).txt ...
Checking file test.c ...
File test.txt is locked.Waiting for unlock message..
File test.txt is now unlocked.
Done sending file test.txt.Total bytes sent 267
```