

## Προγραμματισμός Συστήματος – Εργασία 3

Αθανασία Τουρνάκη  
AM: 1115201600172

Σε αυτή την εργασία υλοποιήθηκαν όλα τα απαιτούμενα. Δημιουργήθηκαν οι εφαρμογές `dropbox_client` και `dropbox_server` με όλες τις λειτουργίες που αναφέρονται στην εκφώνηση.

### Σημαντικές σημειώσεις:

- Για την επικοινωνία μεταξύ του server και των clients χρησιμοποιήθηκαν sockets με τη χρήση της συνάρτησης **`select()`**.
- Μόνο οι διευθύνσεις IP και τα Ports στέλνονται σε binary μορφή μέσω των sockets. Οποιαδήποτε άλλη πληροφορία ανταλλάσσεται σε μορφή **`string`**.
- Το πεδίο **`version`** που χρησιμοποιείται σε πολλές συναρτήσεις και πρωτόκολλά αρχικοποιείται με το **`timestamp της τελευταίας επεξεργασίας του αρχείου`** στο οποίο αναφέρεται και οι συναρτήσεις έχουν τροποποιηθεί αναλόγως.
- Οι συναρτήσεις επεξεργασίας φακέλων και αποστολής αρχείων στηρίζονται κατά πολύ σε αυτές που είχαν δημιουργηθεί στα πλαίσια της **προηγούμενης εργασίας**.
- Υλοποιήθηκε δομή λίστας για την διαχείριση clients τόσο από τον server, όσο και για τους άλλους clients.
- Υλοποιήθηκε δομή κυκλικού buffer για την αποθήκευση των αντικειμένων προς επεξεργασία από τα threads.
- Κατά την εκτέλεση του προγράμματος δημιουργείται για κάθε client ένας mirror φάκελος με το όνομο **`<hostname του client>_mirror`** και για κάθε πελάτη με τον οποίο ανταλλάσσει αρχεία δημιουργείται μέσα σε αυτόν τον φάκελο ένας καινούριος με όνομα **`<hostname καινούριου πελάτη>_<port πελάτη>`**, όπου και αντιγράφονται τελικά τα αρχεία.
- Συμπεριλαμβάνονται μαζί με τον κώδικα, δυο **scripts**.
  - Το **`create_infiles.sh`** είναι ίδιο με αυτό που είχε ζητηθεί στην προηγούμενη εργασία και μπορεί να χρησιμοποιηθεί για να αρχικοποιήσει τον φάκελο `dirName` στον οποίο ο κάθε client κρατάει τα αρχεία του.
  - Η εκτέλεση του script έχει ως εξής:  
**`./create_infiles.sh dirName number_of_files number_of_directories_levels`**.
  - Το **`remove-directories.sh`** μπορεί να χρησιμοποιηθεί για να διαγραφούν οι φάκελοι που περιέχουν τη λέξη **`mirror`**, καθώς και αυτοί που περιέχουν τη λέξη **`input`**.
- Για να εκτελεστεί σωστά το πρόγραμμα πρέπει **απαραίτητα να υπάρχει στον ίδιο φάκελο ο φάκελος `input` “dirName” που δίνεται σαν όρισμα και να μην υπάρχει ο φάκελος `mirror`**.

- Για την επεξεργασία των strings που διαμοιράζονται μεταξύ των threads, χρησιμοποιήθηκαν **thead safe** συναρτήσεις. Πχ, αντί για την strtok χρησιμοποιήθηκε η strtok\_r.

- Για την διαχείριση του σήματος SIGINT(και SIGQUIT) χρησιμοποιήθηκε **signal handler** ο οποίος αλλάζει την τιμή μίας global μεταβλητής terminate την οποία μοιράζονται το main thread και όλα τα υπόλοιπα. Όταν διακοπεί η εκτέλεση του dropbox\_client το main thread κάνει cancel στα υπόλοιπα threads. Εσωτερικά των threads έχει οριστεί μια συνάρτηση **cleanup\_handler** που καλείται μόλις ένα thread γίνει cancel. Αυτή το βοηθάει να ξεκολλήσει από onditional waits ή αναμονή αποδέσμευσης κάποιου mutex και να τερματίσει χωρίς leaks.

### ΠΗΓΕΣ:

- Η δομή circularBuffer και η υλοποίησή της είναι στηριγμένη σε μεγάλο βαθμό στις διαφάνειες και τον κώδικα του τμήματος περιπτώσεων sdi του μαθήματος.

- Για την ανάκτηση πληροφοριών δικτύου χρησιμοποιήθηκαν έτοιμες συναρτήσεις από τη σελίδα GeeksForGeeks, για την οποία είχε δοθεί link στο Piazza σε κάποια σχετική ερώτηση.

### Οδηγίες για την εκτέλεση:

Έχει δημιουργηθεί makefile, οπότε ο χρήστης καλείται μόνο να πληκτρολογήσει make για την μεταγλώττιση του κώδικα και make clean για την απομάκρυνση των object files και των εκτελέσιμων αργότερα.

### Εκτέλεση

Για τον server:

**./dropbox\_server -p portNum**

Για τους clients:

**./dropbox\_client -d dirName -p portNum -w workerThreads -b bufferSize -sp serverPort -sip serverIP**

Θα ήταν προτιμότερο να μπει ./ μπροστά από το όνομα του **dirName**.