

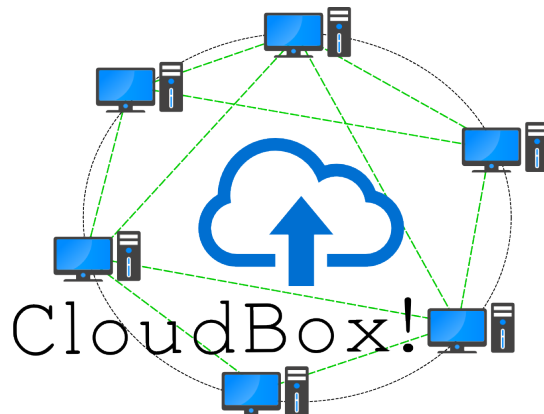
CloudBox!: Ένα εργαλείο cloud αποθήκευσης αρχείων με κατακευματισμένο τρόπο

Project Phase 1
HY - 335α

Χειμερινό εξάμηνο 2012-2013

Διδάσκουσα: Παπαδοπούλη Μαρία
Υπεύθυνος βοηθός: Σουρλίγκας Μανώλης,
surligas@csd.uoc.gr

23 Δεκεμβρίου 2013



Παράδοση πρώτης φάσης: 20/1/2014

Περίληψη

Στην πρώτη φάση του project ο κάθε Cloudbox client θα πρέπει να είναι σε θέση να αναγνωρίζει τις αλλαγές των αρχείων του directory που παρακολουθεί και να ενημερώνει κατάλληλα όλους τους άλλους clients που τρέχουν στο ίδιο subnet, στέλνοντας ένα broadcast μήνυμα χρησιμοποιώντας UDP. Κάθε client που λαμβάνει ένα broadcast μήνυμα, θα ελέγχει την κατάσταση του δικού του directory και αν χρειάζονται να γίνουν αλλαγές, θα συνδέεται στον κατάλληλο client χρησιμοποιώντας TCP, για να ξεκινήσει η μεταφορά του αρχείου.

1 Γενική μορφή μηνυμάτων

Για να μπορούν οι clients να επικοινωνήσουν θα πρέπει το κάθε μήνυμα να έχει μια συγκεκριμένη μορφή. Τα πρώτα 2 bytes του μηνύματος περιέχουν το είδος του μηνύματος. Οι διάφοροι τύποι μηνυμάτων είναι οι εξής:

2 First Bytes	Type
0x01	Μήνυμα Κατάστασης
0x02	Το directory του client δεν έχει αλλαγές
0x03	Ένα νέο αρχείο προστέθηκε
0x04	Ένα αρχείο τροποποιήθηκε
0x05	Ένα αρχείο διαγράφηκε
0x06	Ένας άλλος client ζητάει ένα αρχείο
0x07	Ένας άλλος client προσφέρει ένα αρχείο
0x08	Το directory δεν περιέχει κάποιο αρχείο
0xFFFF	Καμία ενέργεια. Χρησιμοποιείται για debug.

Φυσικά μπορείτε να επεκτείνετε τους πιθανούς τύπους μηνυμάτων και να προσθέσετε αυτούς που σας βολεύουν.

Τον τύπο του μηνύματος ακολουθούν μια σειρά από άλλα fields, άλλα υποχρεωτικά και άλλα υποχρεωτικά ανά περίπτωση. Θα πρέπει να τοποθετούνται με την σειρά που αναφέρονται παρακάτω:

- **Client name:** Μεταβλητό μήκος. Για να μπορέσει να αναγνωρίσει ο receiver το όνομα του sender, το όνομα τοποθετείται ανάμεσα στα bytes 0x0 και 0x0. Αυτό γίνεται γιατί το 0x0 δεν είναι printable χαρακτήρας, οπότε είναι αδύνατο να δόθηκε σαν client name. Υποχρεωτικό για κάθε μήνυμα.
- **Client TCP listening port:** Μήκος 2 bytes. Περιέχει την port στην οποία περιμένει ο client για TCP connections. Η port αυτή πρέπει να είναι κάποια τυχαία διαθέσιμη που παρέχει το λειτουργικό και όχι κάποια ορισμένη από τον χρήστη. Υποχρεωτικό για κάθε πακέτο.
- **Current time timestamp:** Μήκος 8 bytes. Περιέχει την ώρα που στάλθηκε το πακέτο σε δευτερόλεπτα από το Epoch time. Υποχρεωτικό για κάθε μήνυμα.
- **File modification timestamp:** Μήκος 8 bytes. Περιέχει την ώρα που τροποποιήθηκε το αρχείο σε δευτερόλεπτα από το Epoch time. Υποχρεωτικό ανά περίπτωση.
- **File name:** Το όνομα του αρχείου. Μεταβλητό μήκος. Για να μπορέσει να αναγνωρίσει ο receiver το όνομα του αρχείου, το όνομα τοποθετείται ανάμεσα στα bytes 0x0 και 0x0. Αυτό γίνεται γιατί το 0x0 δεν είναι printable χαρακτήρας, οπότε είναι αδύνατο να δόθηκε σαν όνομα αρχείου. Υποχρεωτικό ανά περίπτωση.
- **SHA1 checksum:** Μήκος 20 bytes. Το SHA1 checksum του αρχείου ή το SHA1 όλων των επιμέρους SHA1 των αρχείων, στην περίπτωση που το directory δεν έχει αλλαγές από την τελευταία φορά που ελέγχθηκε. Υποχρεωτικό ανά περίπτωση.
- **File length:** Μήκος 8 bytes. Το μέγεθος του πακέτου σε bytes. Υποχρεωτικό ανά περίπτωση.

Παράδειγμα ενός μηνύματος που δηλώνει ότι το directory δεν έχει αλλαγές:

2 bytes	Variable Length	2 bytes	8 bytes	20 bytes
0x02	0x0test10x0	1881	1548784512	29b12....1e3f5

Το πρόγραμμα χρησιμοποιεί δύο ειδών πρωτόκολλα για τα μηνύματα.

- **UDP:** Ο client στέλνει ένα UDP broadcast πακέτο κάθε φορά που ελέγχει το directory για αλλαγές αρχείων. Ανάλογα με την κατάσταση του directory κάνει broadcast αν αυτό είναι άδειο, αν δεν άλλαξε κανένα αρχείο από την τελευταία φορά που ελέγχθηκε ή αν κάποιο αρχείο προστέθηκε ή αφαιρέθηκε.
- **TCP:** Ένας client συνδέεται χρησιμοποιώντας TCP με έναν άλλον, για να στείλει ή λάβει ένα αρχείο καθώς και (προαιρετικά) για να επιβεβαιώσει την διαγραφή ενός άλλου.

2 Παρακολούθηση για αλλαγές αρχείων στον τοπικό φάκελο

Βασική λειτουργία του προγράμματος είναι ο περιοδικός έλεγχος του directory για τυχόν αλλαγές. Αυτό μπορεί να γίνει εύκολα κρατώντας χαρακτηριστικές ιδιότητες των αρχείων σε μια λίστα. Η struct της λίστας ορίζεται στο cloudbox.h header file και περιέχει τα εξής πεδία.

```
struct dir_files_status_list {  
    char *filename;  
    size_t size_in_bytes;  
    char sha1sum[SHA1_BYTES_LEN];  
    long long int modification_time_from_epoch;  
  
    struct dir_files_status_list *previous;  
    struct dir_files_status_list *next;  
};
```

Το filename, το μέγεθος του αρχείου, το SHA1 checksum και η ώρα τροποποίησης είναι αρκετά ώστε να μπορεί το πρόγραμμα να αναγνωρίσει αν έγινε ή όχι κάποια αλλαγή στο directory. Μετά από κάθε έλεγχο για αλλαγές, ο client στέλνει ένα UDP broadcast μήνυμα, αναφέροντας την κατάσταση του directory του σε όλους

τους client του subnet χρησιμοποιώντας τον κατάλληλο τύπο μηνύματος.

Οι βασικές περιπτώσεις είναι οι εξής:

1. **To directory δεν περιέχει αρχεία.** Σε αυτή την περίπτωση το μήνυμα έχει τύπο 0x08 και εκτός των υποχρεωτικών fields περιέχει την ώρα του client και την ώρα τροποποίησης του τελευταίου αρχείου πριν το directory αδειάσει ή 0 αν αυτό δεν υπήρχε.
2. **To directory περιέχει αρχεία, αλλά δεν έγιναν αλλαγές από την τελευταία προσπέλαση.** Σε αυτή την περίπτωση το μήνυμα έχει τύπο 0x2. Εκτός των υποχρεωτικών fields περιέχει την ώρα του client, την πιο πρόσφατη ώρα τροποποίησης αρχείου, καθώς και το SHA1 του string που σχηματίζεται από τα SHA1 checksums όλων των αρχείων, αύξουσα ταξινομημένα με βάση το όνομά τους. Αυτό γίνεται, διότι κάποια στιγμή μπορεί να εισέλθει ένας client με "clean" directory το οποίο όμως είναι διαφορετικό από το "clean" directory ενός άλλου client. Επομένως οι δύο clients θα πρέπει να συγχρονίσουν το directory τους με τα πιο πρόσφατα αρχεία.
3. **Ένα αρχείο προστέθηκε.** Το μήνυμα έχει τύπο 0x03 και εκτός των υποχρεωτικών fields περιλαμβάνει την ώρα του client, την ώρα τροποποίησης του αρχείου, το όνομά του, το μέγεθός του καθώς και το SHA1 checksum του.
4. **Ένα αρχείο τροποποιήθηκε.** Το μήνυμα έχει τύπο 0x04 και τα ίδια fields με την προηγούμενη περίπτωση.
5. **Ένα αρχείο διαγράφηκε.** Το μήνυμα έχει τύπο 0x05 και τα ίδια fields με τις προηγούμενες δύο περιπτώσεις.

3 Παρακολούθηση για αλλαγές αρχείων από άλλους Cloudbox Clients

Κάθε client παρακολουθεί το δίκτυο για broadcast μηνύματα από άλλους clients και ενεργεί κατάλληλα για κάθε τύπο μηνύματος.

Για αυτή την φάση, θα πρέπει ο client να είναι σε θέση να λαμβάνει αυτά τα broadcast μηνύματα, να εκτυπώνει ένα μήνυμα που να εμφανίζει:

1. Το όνομα του client
2. Τον τύπο του μηνύματος
3. Την IP του remote client την οποία θα πρέπει να βρείτε **προγραμματιστικά** και όχι μέσω κάποιου field στα περιεχόμενα του μηνύματος
4. Την TCP listening port του remote client
5. Ό,τι άλλο κρίνετε απαραίτητο

Επίσης θα πρέπει να στέλνει στον client από τον οποίο έλαβε το broadcast μήνυμα, ένα debug μήνυμα χρησιμοποιώντας TCP, το οποίο με την σειρά του θα εκτυπώνεται στον client που το λαμβάνει.

4 Command Line Arguments

Σας δίνετε το clientbox.c αρχείο το οποίο περιέχει μια main. Τα arguments της main δεν μπορείτε να τα αλλάξετε. Μπορείτε να προσθέσετε βέβαια τα δικά σας.

Πιο αναλυτικά οι παράμετροι της main είναι οι εξής:

- **-n clientname:** Το όνομα του client. Θα πρέπει να είναι μοναδικό για το subnet στο οποίο τρέχει ο cloudbox client.

- **-d directory:** Το path του directory του οποίου τα αρχεία θέλουμε να συγχρονίσουμε. Αν ο χρήστης δώσει κάποιο relative path ή symbolic links αυτό θα πρέπει να μετατραπεί σε absolute path, για την αποφυγή τυχόν προβλημάτων.
- **-i interval:** Το όρισμα αυτό ορίζει κάθε πόσα λεπτά ο client θα ελέγχει για αλλαγές στα αρχεία που περιέχει το directory που παρακολουθεί.
- **-b port:** Η πόρτα την οποία θα χρησιμοποιήσει ο client για να στείλει και να λάβει broadcast UDP μηνύματα. Προφανώς όλοι οι clients θα πρέπει να χρησιμοποιούν την ίδια για να είναι σε θέση να επικοινωνήσουν.

5 Άλλες πληροφορίες

Για να είστε σε θέση να χρησιμοποιήσετε την SHA1() συνάρτηση και να κάνετε compile το πρόγραμμα θα πρέπει να έχετε εγκατεστημένα στον υπολογιστή σας τα development (header) files της openssl βιβλιοθήκης. Το Makefile που σας παρέχεται βάζει τα κατάλληλα flags, οπότε δεν χρειάζεται να κάνετε οτιδήποτε άλλο.

Επίσης θα πρέπει να υλοποιήσετε όλες τις συναρτήσεις που βρίσκονται στο header file και να προσθέσετε τις δικές σας. Συναρτήσεις που καταλήγουν σε `_thread()` θα πρέπει να τρέχουν σε ξεχωριστά threads.

Η πρώτη φάση θα έχει συμβουλευτικό χαρακτήρα, δεν θα βαθμολογηθεί αλλά θα είναι pass or fail. Βασικές απαιτήσεις αυτής της φάσης είναι η στοιχειώδης παρακολούθηση του directory για αλλαγές, η αποστολή και λήψη UDP μηνυμάτων και η αποστολή και λήψη debug TCP μηνύματος.

Η εξέταση θα γίνει σε Linux μηχανήματα, οπότε το project σας θα πρέπει να τρέχει σε αυτό το περιβάλλον. Αν δεν έχετε πολλούς υπολογιστές σε ένα δίκτυο, μπορείτε να χρησιμοποιήσετε εικονικά μηχανήματα που παρέχονται σε φοιτητές από το ΕΔΕΤ. Για περισσότερες πληροφορίες <https://vima.grnet.gr>