

ΥΛΟΠΟΙΗΣΗ Setup

Πίνακας περιεχομένων

ΕΚΤΟΣ ΑΠΟ ΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ	1
Συνοψίζοντας	3
Παραδείγματα υλοποίησης:	3

ΕΚΤΟΣ ΑΠΟ ΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ

1. Περιγραφή του συστήματος και των λειτουργιών του:

- Το σύστημα αποτελείται από δύο κύριες εφαρμογές οι οποίες χωρίζονται σε 4-5 κλάσεις ώστε ο κωδικός να είναι πιο ορανομένος και κατανοητός:
 - α) Streaming Server:**
 - Διαχειρίζεται μια λίστα των διαθέσιμων αρχείων βίντεο. (VideoManager.class)
 - Ελέγχει αν υπάρχουν όλες οι απαιτούμενες μετατροπές (σε διαφορετικές αναλύσεις και μορφές) για κάθε βίντεο και, εάν υπάρχουν κενά, ενεργοποιεί τη μετατροπή τους μέσω του FFmpeg.
 - Επικοινωνεί με τον πελάτη (Client) μέσω Java Sockets, αποστέλλοντας τις πληροφορίες των αρχείων και ξεκινώντας τη ροή του βίντεο.(ClientHandler.class)
 - β) Streaming Client:**
 - Συνδέεται με τον Streaming Server για να λαμβάνει τη λίστα των διαθέσιμων βίντεο.
 - Επιτρέπει στον χρήστη να επιλέξει το επιθυμητό βίντεο και το πρωτόκολλο μετάδοσης.
 - Με βάση τη σύνδεση (και ενδεχομένως με χρήση βιβλιοθήκης για ταχύτητα σύνδεσης) προσαρμόζει την ποιότητα του βίντεο που θα παραληφθεί.
 - Στην τρέχουσα έκδοση η διεπαφή είναι γραμμής εντολών, όμως στις επεκτάσεις μπορεί να υλοποιηθεί γραφικό περιβάλλον (GUI).

2. Παρουσίαση του τελικού υλοποιημένου συστήματος:

- Επικοινωνία Client-Server μέσω Java Sockets, με δυνατότητα εξυπηρέτησης πολλαπλών πελατών ταυτόχρονα (multi-threading).
- Αυτόματη αναγνώριση των αρχείων βίντεο στον κατάλογο και ανάλυση/μετατροπή τους σε διαφορετικές αναλύσεις και μορφές με χρήση FFmpeg (μέσω του Jaffree).
- Καταγραφή σημαντικών γεγονότων και λαθών κατά τη διαδικασία μετατροπής (μέσω καταγραφής στο console, με προτάσεις για μελλοντική ενσωμάτωση Logger).
- Σενάρια εκτέλεσης που περιλαμβάνουν:
 - Την εκκίνηση του Server και τη λήψη του αρχικού μηνύματος συστήματος.
 - Την επικοινωνία του Client για την αίτηση λίστας βίντεο και την λήψη των σχετικών πληροφοριών.
 - Παραδείγματα εκτέλεσης όπου ο χρήστης πληκτρολογεί εντολές LIST, GET κλπ.

3. Παρουσίαση των προβλημάτων που αντιμετωπίσατε και πώς αυτά επιλύθηκαν (ή έμειναν ανοικτά):

- Διαχείριση ταυτόχρονων προσβάσεων κατά την εκτέλεση μετατροπών βίντεο:

Εφαρμόστηκε σύστημα κλειδώματος αρχείων (File Lock) για αποφυγή διπλών ενεργειών.

- Επικοινωνία Client-Server: Αρχικά προέκυψαν ζητήματα στην σύνδεση και στην ομαλή διαχείριση πολλαπλών συνδέσεων, για τα οποία εφαρμοστεί multi-threading και χρήση shutdown hook για ομαλή διακοπή.
- Ενσωμάτωση του FFmpeg μέσω του Jaffree: Εμφανίστηκαν προβλήματα αναγνώρισης παραμέτρων για τη μετατροπή και απαιτήθηκε η προσαρμογή των παραμέτρων μετατροπής (bitrate, resolution).
- Ορισμένες λειτουργίες (όπως η δυναμική αλλαγή ποιότητας με βάση το bandwidth και η ενσωμάτωση GUI) παραμένουν σε στάδιο μελλοντικής ανάπτυξης.

4. Βελτίωση της λειτουργίας απελευθέρωσης κλειδώματος:

- Πριν την υλοποίηση της βοηθητικής κλάσης (LockWithChannel), παρατηρήθηκε ότι κάποια αρχεία κλειδώματος δεν διαγράφονταν μετά την επιτυχή δημιουργία του βίντεο. Αν προέκυπτε οποιοδήποτε σφάλμα ή αν το αρχείο είχε καταλήξει σε ασυνεπή κατάσταση, παρέμενε στον κατάλογο, εμποδίζοντας νέες μετατροπές.
- Με τη δημιουργία της βοηθητικής κλάσης και τον έλεγχο μεταβλητών: Δημιουργήθηκε η κλάση LockWithChannel που αποθηκεύει το FileLock, το FileChannel και το RandomAccessFile που προκύπτουν κατά την απόκτηση του κλειδώματος. Όλες οι μεταβλητές ελέγχονται εντός try-catch block, ώστε σε περίπτωση εξαιρέσεων να εξασφαλίζεται ο σωστός τερματισμός (απελευθέρωση πόρων και κλείσιμο Streams).
- Συνοπτικά, ο ενιαίος έλεγχος μέσα σε try-catch διασφαλίζει ότι κάθε στοιχείο της βοηθητικής κλάσης είναι σωστά ελεγχόμενο και απελευθερωμένο, διορθώνοντας τα αρχικά σφάλματα διαχείρισης πόρων.

Συνοψίζοντας

- Το σύστημα υλοποιεί τις βασικές απαιτήσεις του έργου αλλά αφήνει περιθώρια για επεκτάσεις, όπως:
- την υποστήριξη διαφορετικών πρωτοκόλλων μετάδοσης (UDP, TCP, RTP/UDP),
- την υλοποίηση γραφικού περιβάλλοντος, και
- την ενσωμάτωση συστήματος κρυπτογράφησης για ασφαλή επικοινωνία.

Παραδείγματα υλοποίησης:

1. **Παράδειγμα εκκίνησης του Server:** *"Server running on port 5058"* Υλοποιείται ως εξής:

```
[source,java]
----
System.out.println("Server running on port " + port);
----
```

2. **Παράδειγμα διαχείρισης πολλαπλών συνδέσεων:** *Χρήση του AtomicInteger:*

```
[source,java]
----
if (activeConnections.get() >= MAX_CONNECTIONS) {
    Thread.sleep(100);
    continue;
}
----
```

3. **Παράδειγμα χρήσης του File Lock στην μετατροπή βίντεο:**

```
[source,java]
----
if (!acquireLock(videoName, targetQuality, targetFormat)) {
    System.out.println("Conversion already in progress by another instance: " +
        videoName + "-" + targetQuality + "." + targetFormat);
    return;
}
----
```

4. **Παράδειγμα επεξεργασίας εντολών από τον Client:**

```
[source,java]
----
if (command.startsWith("LIST")) {
    return videoManager.getVideoList();
}
-----
```

```
} else if (command.startsWith("GET ")) {  
    String videoName = command.substring(4);  
    return videoManager.getVideoInfo(videoName);  
}  
----
```