

Δίκτυα Υπολογιστών : Project Μέρος 1

Κωνσταντίνα Σουβατζιδάκη, 3170149

Λυδία Αθανασίου, 3170003

2020-21

1. Σύντομη Ανάλυση της Υλοποίησης:

Η υλοποίηση αποτελείται από 2 κλάσεις, τις Peer και Tracker.

Tracker:

Ο Tracker περιέχει τις εξής δομές δεδομένων, οι οποίες υποστηρίζουν ταυτόχρονη επεξεργασία από νήματα (ConcurrentHashMap και Collections.synchronizedList):

- **registered_peers:** κλειδί: username, τιμή: (password, count_downloads, count_failures)
- **loggedin_peers:** κλειδί: token_id, τιμή: (ip_address, port, user_name)
- **file_names:** λίστα με όλα τα ονόματα αρχείων στο σύστημα
- **files_peers:** κλειδί: file name, τιμή: λίστα όλων των token_ids των peers που περιέχουν το αρχείο

Η κλάση Tracker αποτελείται από τις εξής μεθόδους, που υλοποιούν τις λειτουργίες του Tracker όπως αυτές περιγράφονται στην εκφώνηση της εργασίας.

- **accept_requests():** Μέσω ενός ServerSocket, δέχεται αιτήματα από τους peers, για κάθε αίτημα κατασκευάζει ένα νέο νήμα (Thread της Java) και ανάλογα με τον τύπο του αιτήματος, καλεί την αντίστοιχη μέθοδο που εξυπηρετεί του συγκεκριμένου τύπου αιτήματα.
- **register(input, output):** Η συγκεκριμένη μέθοδος υλοποιεί την απόκριση σε αιτήματα εγγραφής (register) νέου Peer στο σύστημα. Δέχεται μηνύματα από τον peer μέσω του ObjectInputStream input και στέλνει αποκρίσεις μέσω του ObjectOutputStream output. Πιο συγκεκριμένα, δέχεται από τον peer ένα username, και τον ενημερώνει κατάλληλα εάν το username αυτό είναι διαθέσιμο ή όχι (κάθε username πρέπει να είναι μοναδικό). Αυτό συνεχίζεται μέχρι ο peer να στείλει ένα μοναδικό username. Ύστερα ζητείται ένας κωδικός, και τα στοιχεία του peer αποθηκεύονται στην δομή registered_peers.
- **login_and_inform(input, output):** Η συγκεκριμένη μέθοδος υλοποιεί την απόκριση σε αιτήματα σύνδεσης (login) ενός εγγεγραμμένου Peer στο σύστημα. Αρχικά, ταυτοποιεί τον peer, λαμβάνοντας username και passwords έως ότου να δοθούν αυτά που να αντιστοιχούν με τα αποθηκευμένα στην δομή registered_peers. Ο συνδεδεμένος peer στέλνει ύστερα την λίστα με τα αρχεία του (λειτουργία inform), και ο Tracker ενημερώνει κατάλληλα τις δομές file_names και files_peers. Για κάθε peer που συνδέεται του δίνεται ένα μοναδικό session token (random) και αποθηκεύονται τα στοιχεία του στην δομή loggedin_peers. Όλοι οι υπόλοιποι συνδεδεμένοι peers ενημερώνονται με ένα μήνυμα “update” ώστε να ζητήσουν εκ νέου την λίστα των αρχείων.
- **logout(input, output):** Η συγκεκριμένη μέθοδος υλοποιεί την απόκριση σε αιτήματα αποσύνδεσης (logout) ενός εγγεγραμμένου Peer στο σύστημα. Ο Tracker αποδεδμεύει το token του peer αφαιρώντας το από τις δομές loggedin_peers και files_peers.

- **reply_list(output):** Η συγκεκριμένη μέθοδος υλοποιεί την απόκριση σε αιτήματα list από Peers στο σύστημα. Στέλνει στον peer την δομή file_names με όλα τα ονόματα αρχείων στο σύστημα.
- **reply_details(input, output):** Η συγκεκριμένη μέθοδος υλοποιεί την απόκριση σε αιτήματα details από Peers. Πιο συγκεκριμένα, αρχικά ελέγχει αν το αρχείο υπάρχει στο σύστημα και στέλνει ανάλογη απόκριση. Εάν υπάρχει, βρίσκει τα token των peers που το περιέχουν μέσω της δομής files_peers. Για κάθε peer ελέγχει αν είναι ενεργός στέλνοντάς του αίτημα “checkactive”. Επιστρέφει τις λεπτομέρειες, που βρίσκονται αποθηκευμένες στις δομές του Tracker, μόνο των peers που απαντούν θετικά στο αίτημα αυτό. Για όσους peers δεν απαντήσουν στο αίτημα “checkactive”, αποδεδεσμεύει το token τους αφαιρώντας το από τις αντίστοιχες δομές.
- **reply_notify(input, output):** Η συγκεκριμένη μέθοδος υλοποιεί την απόκριση σε αιτήματα notify από Peers στο σύστημα. Συγκεκριμένα, ανάλογα με το αν η ενημέρωση είναι θετική ή αρνητική, ενημερώνει κατάλληλα τις δομές του για τους εμπλεκόμενους Peers.

Peer:

Ο Peer περιέχει μια δομή δεδομένων που περιλαμβάνει τα ονόματα των αρχείων στο shared directory του, και μία ακόμα που περιλαμβάνει τα ονόματα όλων των αρχείων στο σύστημα.

Η κλάση Peer αποτελείται από τις εξής μεθόδους, που υλοποιούν τις λειτουργίες του Peer όπως αυτές περιγράφονται στην εκφώνηση της εργασίας.

- **init():** Η συγκεκριμένη μέθοδος αρχικοποιεί την δομή με τα αρχεία του shared directory.
- **connect():** Η συγκεκριμένη μέθοδος δίνει στον Peer την επιλογή να επιλέξει αν θέλει να εγγραφεί ή να συνδεθεί εάν είναι ήδη εγγεγραμμένος.
- **register():** Η συγκεκριμένη μέθοδος υλοποιεί την αποστολή αιτημάτων εγγραφής (register) νέου Peer στο σύστημα. Λειτουργεί συνεργατικά με την αντίστοιχη μέθοδο register() του Tracker, όπως περιγράφεται παραπάνω. Μόλις ένας Peer εγγραφεί, απευθείας οδηγείται στην λειτουργία login.
- **login_and_inform():** Η συγκεκριμένη μέθοδος υλοποιεί την αποστολή αιτημάτων σύνδεσης (login) ενός εγγεγραμμένου Peer στο σύστημα. Λειτουργεί συνεργατικά με την αντίστοιχη μέθοδο login_and_inform() του Tracker, όπως περιγράφεται παραπάνω.
- **start():** Η συγκεκριμένη μέθοδος, αφού συνδεθεί ο Peer στο σύστημα, δημιουργεί 2 νήματα, ώστε ο Peer να μπορεί παράλληλα τόσο να στέλνει όσο και να δέχεται αιτήματα.
 - Το πρώτο νήμα, μέσω ενός ServerSocket, δέχεται αιτήματα από τον Tracker και τους υπόλοιπους Peers. Για κάθε αίτημα δημιουργεί ένα ξεχωριστό νήμα που καλεί την αντίστοιχη μέθοδο για την εξυπηρέτησή του. Οι τύποι αιτημάτων είναι “checkactive”, “file”, “update”. Όταν ο Peer λάβει αίτημα “update”, ζητά εκ νέου την λίστα με όλα τα αρχεία του συστήματος.
 - Το δεύτερο νήμα δίνει στον Peer την επιλογή είτε να κατεβάσει ένα νέο αρχείο, είτε να κάνει logout. Κατά την δημιουργία του, καλείται αρχικά η μέθοδος list(). Αν ο peer επιλέξει να κατεβάσει αρχείο, του εμφανίζονται όλα τα διαθέσιμα αρχεία του συστήματος. Αν επιλέξει να κατεβάσει ένα αρχείο που ήδη έχει, τότε εμφανίζεται κατάλληλο μήνυμα.

- **logout():** Η συγκεκριμένη μέθοδος υλοποιεί την αποστολή αιτημάτων αποσύνδεσης (login) ενός συνδεδεμένου Peer στο σύστημα. Λειτουργεί συνεργατικά με την αντίστοιχη μέθοδο logout() του Tracker, όπως περιγράφεται παραπάνω, στέλνοντας του το token_id του.
- **list():** Η συγκεκριμένη μέθοδος υλοποιεί την αποστολή αιτημάτων list. Λαμβάνει από τον Tracker την δομή file_names με όλα τα ονόματα αρχείων στο σύστημα.
- **details(filename):** Η συγκεκριμένη μέθοδος υλοποιεί την αποστολή αιτημάτων για τις λεπτομέρειες των peers που διαθέτουν το αρχείο με όνομα filename. Ο Tracker αποκρίνεται σε αυτά τα αιτήματα μέσω της αντίστοιχης μεθόδους reply_details(), όπως περιεγράφηκε παραπάνω.
- **reply_simpleDownload(input, output):** Η συγκεκριμένη μέθοδος υλοποιεί την απόκριση σε αιτήματα για κατέβασμα αρχείων από άλλους Peers. Εάν το αρχείο υπάρχει στον peer, το διαβάζει και το στέλνει σε μορφή bytes μέσω του ObjectOutputStream output. Διαφορετικά στέλνει μήνυμα πως το αρχείο δεν υπάρχει.
- **checkactive(output):** Η συγκεκριμένη μέθοδος υλοποιεί την απόκριση σε αιτήματα “checkactive”
- **simpleDownload(details_reply, file):** Η συγκεκριμένη μέθοδος υλοποιεί την αποστολή αιτημάτων για κατέβασμα αρχείων από άλλους Peers. Ο Peer, μέσα από την δομή με τις λεπτομέρειες των υπόλοιπων Peers που λαμβάνει προηγουμένως από την λειτουργία details, ελέγχει τον κάθε Peer αν είναι ενεργός στέλνοντας του αίτημα “checkactive”, και επιλέγει τον καλύτερο ανάλογα με τον χρόνο απόκρισης και τις μετρικές count_downloads και count_failures. Εάν για κάποιο λόγο δεν σταλεί το αρχείο, ζητείται από τον αμέσως επόμενο Peer. Η ταξινόμηση των peers με βάση τις παραπάνω μετρικές γίνεται με ευκολία με χρήση δομής TreeMap της Java.
- **send_notify(file_sent, peer_username, sent):** Η συγκεκριμένη μέθοδος υλοποιεί την αποστολή ενημέρωσης για την πορεία ενός download αρχείου στον Tracker. Ανάλογα με το αν το download ήταν επιτυχές ή όχι, ο Tracker ενημερώνει τις δομές του κατάλληλα, όπως περιεγράφηκε παραπάνω στην αντίστοιχη μέθοδο reply_notify.

2. Περιγραφή Αρχείων:

Η εφαρμογή αποτελείται από 2 αρχεία Java, τα Peer.java και Tracker.java, όπου το καθένα αποτελείται από την αντίστοιχη κλάση και από μια μέθοδο main για την εκτέλεση των Trackers και Peers αντίστοιχα. Τα αρχεία που αποστέλλονται μέσω του συστήματος peer2peer είναι τύπου .mp3 και βρίσκονται χωρισμένα σε 5 καταλόγους. Κάθε κατάλογος αντιστοιχεί σε έναν Peer και περιέχει 2-3 αρχεία .mp3.

3. Τρόπος Εκτέλεσης:

Αρχικά, εκτελείται ο Tracker, και ακολουθούν οι Peers (java Peer path_name port), χωρίς να παίζει κάποιον ρόλο η σειρά που θα εκτελεστούν. Ο Tracker καθώς και όλοι οι Peers τρέχουν σε localhost (ip του μηχανήματος) με ένα ξεχωριστό port για το καθένα.

Δεν αντιμετωπίσαμε κάποιο πρόβλημα στην ανάπτυξη της εφαρμογής, το οποίο να μας αναγκάσει να αποκλίνει η υλοποίηση από τις προδιαγραφές της εκφώνησης.

4. Screenshots της εξόδου:

- Register ενός Peer στον Tracker:

```
Welcome to peer. Register or Login?
1) Register
2) Login
0) Exit
1
Registering new peer
Insert username:
Peer1
Username accepted!
Insert password:
peer1
Register was successful!
```

Από την πλευρά του Peer

```
Received a new request from a peer : REGISTER
Username Accepted
Successfully registered peer with Username = Peer1 and Password = peer1
```

Από την πλευρά του Tracker

- Login ενός Peer στον Tracker:

```
Login
Insert username:
Peer1
Username accepted!
Insert password:
peer2
Wrong Password! Try again:
peer1
Password correct! Session token of peer: 438348479

Informed tracker about ip,port and files on this peer!
```

Από την πλευρά του Peer

```
Received a new request from a peer : LOGIN
Wrong Password. Request new.
Password accepted! Login Successful. Token for new peer = 438348479
Received Peer information: ip = 192.168.2.2, port = 6001 Files:
A Waltz For Naseem.mp3
Le Baguette.mp3
All logged-in peers currently:
{438348479=[192.168.2.2, 6001, Peer1]}
```

Από την πλευρά του Tracker

```
Received a new request from a peer or Tracker : UPDATE
Requesting a list of all files in the P2P system. . .
Seccessfully received a list of all files!
```

Από την πλευρά των ήδη συνδεδεμένων Peers

- Logout ενός Peer από τον Tracker:

```
Choose:
1) Select file
0) Logout
0
Logout successful!

Stopped receiving requests
```

Από την πλευρά του Peer

```
Received a new request from a peer : LOGOUT
Logging out from peer with token id = 438348479
```

Από την πλευρά του Tracker

- Επιτυχής ερώτηση για τα διαθέσιμα αρχεία και λήψη της σχετικής απάντησης (λειτουργία list):

```
Requesting a list of all files in the P2P system. . .
Seccessfully received a list of all files!
```

Από την πλευρά του Peer

```
Received a new request from a peer : LIST
List of all files sent successfully!
```

Από την πλευρά του Tracker

- Επιτυχής αναζήτηση για ένα αρχείο και λήψη της σχετικής απάντησης (λειτουργία details).

```

1) Ambient Bongos.mp3
2) Ambush in Rattlesnake Gulch.mp3
3) Apex.mp3
4) A Waltz For Naseem.mp3
5) Le Baguette.mp3
Choose a file. Press any other key to exit. . .
1
Requesting details for file Ambient Bongos.mp3. . .
File exists! Receiving details. . .
Received details. Peers that have the file:
{841486986=[192.168.2.2, 6002, Peer2, 0, 0], 332501323=[192.168.2.2, 6003, Peer3, 0, 0]}

```

Από την πλευρά του Peer που ζητά το αρχείο

```

Received a new request from a peer : DETAILS
Requested details for file Ambient Bongos.mp3
File exists
peer is active
peer is active
Details sent succesfully

```

Από την πλευρά του Tracker

```

Received a new request from a peer or Tracker : CHECKACTIVE
Successfully replied to CHECKACTIVE request!

```

Από την πλευρά των Peers που έχουν το αρχείο

- Ανεπιτυχής αναζήτηση για ένα αρχείο λόγω αστοχίας του μοναδικού peer που είχε το αρχείο:

```

1) A Waltz For Naseem.mp3
2) Le Baguette.mp3
3) Ambient Bongos.mp3
4) Ambush in Rattlesnake Gulch.mp3
5) Apex.mp3
Choose a file. Press any other key to exit. . .
4
Requesting details for file Ambush in Rattlesnake Gulch.mp3. . .
File exists! Receiving details. . .
Received details. Peers that have the file:
{}

File wasn't found in any peer.

```

Από την πλευρά του Peer

```
Received a new request from a peer : DETAILS
Requested details for file Ambush in Rattlesnake Gulch.mp3
File exists
peer is not active
Details sent succesfully
```

Από την πλευρά του Tracker

- Αναζήτηση αρχείου που ήδη υπάρχει στον Peer

```
Choose:
1) Select file
0) Logout
1
1) A Waltz For Naseem.mp3
2) Le Baguette.mp3
3) Ambient Bongos.mp3
4) Ambush in Rattlesnake Gulch.mp3
5) Apex.mp3
Choose a file. Press any other key to exit. . .
1
File already exists in peer
```

Από την πλευρά του Peer

- Επιτυχές download ενός αρχείου, με επιλογή από 3 Peers:

```
Peer with token 841486986 is active and responded in 25. The total factor is 25.0
Peer with token 332501323 is active and responded in 15. The total factor is 15.0

Selected peer with the minimum response time: Peer3
File exists in peer Peer3
File successfully downloaded from peer Peer3
Notifying tracker about file transfer!
Notification sent succesfully!
```

Από την πλευρά του Peer που ζήτησε το αρχείο

```
Received a new request from a peer or Tracker : CHECKACTIVE  
Successfully replied to CHECKACTIVE request!
```

```
Received a new request from a peer or Tracker : FILE  
Requested file: Ambient Bongos.mp3  
File found.  
Successfully read file bytes. Sending....  
Sleeping for a while, to simulate download time. . .  
Sent!
```

Από την πλευρά του Peer που επιλέχθηκε

```
Received a new request from a peer : NOTIFY  
Peer with token 879481593 has successfully received file Ambient  
Bongos.mp3 from peer with username Peer3
```

Από την πλευρά του Tracker

- Ανεπιτυχές download ενός αρχείου και ύστερα επιτυχές download από τον αμέσως επόμενο διαθέσιμο Peer

```
1) A Waltz For Naseem.mp3  
2) Ambient Bongos.mp3  
3) Le Baguette.mp3  
4) Monsters in Hotel.mp3  
5) Ukulele Song.mp3  
6) Video Game Blockbuster.mp3  
Choose a file. Press any other key to exit. . .  
6  
Requesting details for file Video Game Blockbuster.mp3. . .  
File exists! Receiving details. . .  
Received details. Peers that have the file:  
{2107411314=[192.168.2.2, 6004, Peer3, 0, 0], -621889727=[192.168.2.2, 6003, Peer2, 0, 0]}  
  
Peer with token 2107411314 is active and responded in 10. The total factor is 10.0  
Peer with token -621889727 is active and responded in 31. The total factor is 31.0  
  
Selected peer with the minimum response time: Peer3  
File exists in peer Peer3  
File transfer with peer Peer3 failed!  
  
Notifying tracker about file transfer!  
Notification sent succesfully!  
  
Selected peer with the minimum response time: Peer2  
File exists in peer Peer2  
File successfully downloaded from peer Peer2  
Notifying tracker about file transfer!  
Notification sent succesfully!
```

Από την πλευρά του Peer που ζητά το αρχείο


```
Received a new request from a peer or Tracker : CHECKACTIVE  
Successfully replied to CHECKACTIVE request!
```

```
Received a new request from a peer or Tracker : CHECKACTIVE  
Successfully replied to CHECKACTIVE request!
```

```
Received a new request from a peer or Tracker : FILE  
Requested file: Video Game Blockbuster.mp3  
File found.  
Successfully read file bytes. Sending....  
Sleeping for a while, to simulate download time. . .  
PS C:\Users\user\Desktop\P2P-File-Transfer>
```

Από την πλευρά του Peer που επιλέχθηκε αρχικά αλλά απέτυχε

```
Received a new request from a peer or Tracker : CHECKACTIVE  
Successfully replied to CHECKACTIVE request!
```

```
Received a new request from a peer or Tracker : CHECKACTIVE  
Successfully replied to CHECKACTIVE request!
```

```
Received a new request from a peer or Tracker : FILE  
Requested file: Video Game Blockbuster.mp3  
File found.  
Successfully read file bytes. Sending....  
Sleeping for a while, to simulate download time. . .  
Sent!
```

Από την πλευρά του Peer που επιλέχθηκε στην συνέχεια

```
Received a new request from a peer : DETAILS  
Requested details for file Video Game Blockbuster.mp3  
File exists  
peer is active  
peer is active  
Details sent successfully
```

```
Received a new request from a peer : NOTIFY  
Peer with token -1057784556 tried to receive file Video Game Blockbuster.mp3 from  
peer with username Peer3 but failed
```

```
Received a new request from a peer : NOTIFY  
Peer with token -1057784556 has successfully received file Video Game Blockbuster.  
mp3 from peer with username Peer2
```

Από την πλευρά του Tracker

5. Γενικότερη αξιολόγηση:

Κατά την υλοποίηση της εφαρμογής δεν συναντήσαμε κάποια ιδιαίτερη δυσκολία, καθώς είχαμε αρκετή εμπειρία με την γλώσσα προγραμματισμού Java, καθώς και με τα sockets και τον πολυνηματικό προγραμματισμό.

Ιδιαίτερα μας διευκόλυναν οι συγχρονισμένες δομές δεδομένων που παρέχει η Java (ConcurrentHashMap και synchronizedList) καθώς δεν χρειάστηκε να μας απασχολήσουν οι συνθήκες ανταγωνισμού από την παράλληλη πρόσβαση πολλών νημάτων.