

# ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

## Τμήμα Πληροφορικής

### Μάθημα «Δίκτυα Υπολογιστών»

Εαρινό Εξάμηνο 2020-21

---

**Διδάσκων: Γεώργιος Δ. Σταμούλης, Καθηγητής**

*Βοηθοί: Διονύσιος Δαμασιώτης, Ιάκωβος Πιτταράς,  
Δημήτριος Διαμαντόγιαννης, Δημήτριος Καραντούσης*

**Προγραμματιστική Άσκηση Υλοποίησης Κατανεμημένου Συστήματος Ομότιμων (P2P)  
για την ανταλλαγή αρχείων – P-tO-P-a (Φάση 1)**

**Ημερομηνία Ανάθεσης: 1/4/2021**

**Ημερομηνία Παράδοσης: 22/4/2021**

#### **1 Περιεχόμενο**

Εξοικείωση με προγραμματισμό σε περιβάλλον υποδοχών (sockets).

#### **2 Γενικές Πληροφορίες**

Η εργασία θα πρέπει να εκπονηθεί σε ομάδες τριών (3) ατόμων. Για την παράδοση της εργασίας, κάθε ομάδα θα ετοιμάσει μία σύντομη αναφορά με τις απαντήσεις της και με τεκμηρίωση, την οποία θα υποβάλλει ηλεκτρονικά μέσω **e-class**. Για απορίες μπορείτε να απευθυνθείτε στο [pittaras@aueb.gr](mailto:pittaras@aueb.gr). Επίσης, θα ακολουθήσει και προφορική εξέταση της εργασίας στο εργαστήριο, που θα περιλαμβάνει σύντομη περιγραφή της εργασίας, των βημάτων που αναπτύχθηκαν, των δυσκολιών που αντιμετωπίστηκαν κλπ.

#### **3 Εισαγωγή**

Η προγραμματιστική εργασία σε περιβάλλον υποδοχών αποτελεί εισαγωγή στο δικτυακό προγραμματισμό. Ζητούμενο της εργασίας είναι η υλοποίηση ενός απλού κατανεμημένου συστήματος ομότιμων οντοτήτων (Peer-2-Peer , P2P) για την ανταλλαγή αρχείων. Η εργασία θα υλοποιηθεί σε δύο φάσεις, με προοδευτικά αυξανόμενη πολυπλοκότητα. Η παρούσα εκφώνηση αναφέρεται στην Φάση 1.

Το σύστημα αποτελείται από έναν tracker και πολλαπλούς παρόμοιους peers, καθένας από τους οποίους διατηρεί ένα λογαριασμό στον tracker. Κάθε peer έχει έναν φάκελο (shared\_directory) στον οποίο τοποθετεί τα αρχεία που επιθυμεί να μοιράζεται με τους υπόλοιπους peers. Ένας peer, αφού πρώτα συνδεθεί με τον tracker, τον ενημερώνει για τα αρχεία που έχει διαθέσιμα. Όταν ένας peer επιθυμεί να κατεβάσει ένα αρχείο, ζητά και λαμβάνει από τον tracker τις πληροφορίες επικοινωνίας (ip, port και άλλες - βλέπε κατωτέρω) των peers που διαθέτουν το αρχείο αυτό και στη συνέχεια συνδέεται με έναν από αυτούς προκειμένου να το κατεβάσει. Ο tracker καθώς και οι peers είναι σε θέση να διατηρούν πολλαπλές συνδέσεις με άλλους peers και να εξυπηρετούν τα αιτήματά τους ταυτόχρονα.

Η εργασία συστήνεται να υλοποιηθεί στη γλώσσα Java.

## 4 Λειτουργίες του Peer

Κάθε peer εκτελεί τις εξής λειτουργίες:

1. Επικοινωνεί με τον tracker και τους άλλους peers με τη χρήση υποδοχών (sockets).
2. Διατηρεί τοπικά έναν φάκελο (shared\_directory), στον οποίο αποθηκεύονται τα αρχεία που μοιράζεται με άλλους.
3. Διατηρεί ένα σύνολο αρχείων. Κατά την έναρξη θα πρέπει κάθε peer να έχει ήδη αποθηκεύσει τουλάχιστον 2 αρχεία από εκείνα τα ονόματα των οποίων που περιέχονται στο αρχείο fileDownloadList.txt. Κάθε αρχείο θα πρέπει να αποθηκεύεται σε έναν τουλάχιστον peer.
4. Δημιουργεί και διατηρεί ένα λογαριασμό στον tracker (υλοποίηση λειτουργίας **register**, **login** και **logout**).
  - a. **Register**: ο peer επιλέγει ένα username και password και τα στέλνει στον tracker. Στην περίπτωση όπου το username δεν χρησιμοποιείται ήδη από άλλον peer, ο tracker δημιουργεί έναν λογαριασμό και επιστρέφει μήνυμα επιτυχίας στον peer. Διαφορετικά επιστρέφει μήνυμα αποτυχίας δημιουργίας του λογαριασμού και απαίτησης για εισαγωγή νέου username από τον peer.
  - b. **Login**: ο peer εισάγει σε ένα μήνυμα το username και το password και το στέλνει στον tracker. Ο tracker πιστοποιεί τον peer και δημιουργεί ένα τυχαίο token\_id με το οποίο και του απαντά. Το token\_id είναι ένας τυχαίος αριθμός ο οποίος ταυτοποιεί τον peer για το συγκεκριμένο session (δηλαδή μέχρι ο peer να κάνει logout). Στη συνέχεια ο peer ενημερώνει τον tracker (λειτουργία **inform**) για τα τρέχοντα περιεχόμενα του shared\_directory του καθώς και για τις πληροφορίες επικοινωνίας (ip\_address, port).
  - c. **Logout**: ο peer ενημερώνει τον tracker ότι αποχωρεί (ταυτοποιώντας τον εαυτό του με το token\_id). Ο tracker αποδέχεται την αποχώρηση, στέλνει μήνυμα επιτυχούς αποχώρησης στον peer, και ενημερώνει κατάλληλα τις δομές δεδομένων που διατηρεί σχετικά με τους peers και ακυρώνει το token\_id.
5. Υλοποιεί τη λειτουργία **list**, με την οποία ζητά από τον tracker την λίστα με τα ονόματα των αρχείων που είναι διαθέσιμα συνολικά στο σύστημα P2P.

6. Υλοποιεί τη λειτουργία **details**, με την οποία ζητά από τον tracker πληροφορίες για ένα συγκεκριμένο αρχείο, που περιλαμβάνεται στην απάντηση της ανωτέρω, αποστέλλοντας το όνομα του αρχείου. Στη λειτουργία details τελικά ο peer λαμβάνει ως απάντηση:
  - a. μία λίστα με τις πληροφορίες επικοινωνίας (ip\_address, port, user\_name, count\_downloads, count\_failures) των peers που έχουν το συγκεκριμένο αρχείο ή
  - b. μήνυμα αποτυχίας στην περίπτωση που το αρχείο δεν υπάρχει σε κανέναν άλλο peer.
7. Υλοποιεί τη λειτουργία **checkActive** με την οποία ελέγχει αν ένας peer που περιλαμβάνεται στην απάντηση της ανωτέρω ερώτησης είναι όντως «ενεργό». Επίσης πρέπει να απαντά με μήνυμα επιβεβαίωσης στα μηνύματα checkActive που προέρχονται από άλλους peers ή από τον tracker.
8. Υλοποιεί τη λειτουργία **simpleDownload**. Ο peer μετά την εκτέλεση της αναζήτησης επιλέγει τον καλύτερο peer από αυτούς που επέστρεψε η λειτουργία details και του ζητά το αρχείο. Για να επιλέξει τον καλύτερο:
  - a. στέλνει μηνύματα checkActive σε κάθε peer και μετρά τον χρόνο απόκρισης για τον καθένα, και
  - b. στη συνέχεια το πολλαπλασιάζει (για καθένα peer) με τον παράγοντα  
 $0.9^{\text{count\_downloads}} \cdot 1.2^{\text{count\_failures}}$ .Ως καλύτερος θεωρείται ο peer εκείνος με τη μικρότερη τιμή του συνδυασμένου αυτού δείκτη ο οποίος «επιβραβεύει» κάθε peer για τα αρχεία που έχει ήδη «δώσει» επιτυχώς και τον «τιμωρεί» για τις προηγούμενες αποτυχίες του. Μετά την επιτυχή λήψη του αρχείου, ο peer το αποθηκεύει στο shared\_directory και ενημερώνει τον tracker (λειτουργία **notify**) ότι διαθέτει πλέον και αυτός το αρχείο καθώς και για το user\_name του peer από τον οποίο δόθηκε επιτυχώς το αρχείο. Σε περίπτωση που η λήψη αυτή αποτύχει, ο peer ενημερώνει (λειτουργία **notify**) τον tracker για την αποτυχία του simpleDownload καθώς και το user\_name του peer από τον οποίο είχε ζητηθεί το αρχείο και επιλέγει τον 2<sup>ο</sup> καλύτερο peer (ως προς τον προαναφερθέντα συνδυασμένο δείκτη) κτλ. Σε περίπτωση που δεν υπάρχει άλλος peer που διαθέτει το αρχείο αυτό η διαδικασία τερματίζεται και εμφανίζεται μήνυμα σφάλματος στον peer που επιθυμεί να κατεβάσει το αρχείο.
9. Απαντά κατάλληλα σε αιτήσεις για κατέβασμα αρχείων από άλλους peers. Συγκεκριμένα, ο peer διαβάζει το αρχείο που του ζητήθηκε από το shared\_directory του και το στέλνει στον peer που το ζήτησε. Αν δεν διαθέτει το αρχείο αυτό (πχ. γιατί απλώς το ερώτημα του υπεβλήθη εκ παραδρομής) απαντά αρνητικά.

## 5 Λειτουργίες του Tracker

Ο tracker εκτελεί τις εξής λειτουργίες:

1. Διατηρεί κατάλληλη δομή δεδομένων με τις πληροφορίες κάθε εγγεγραμμένου peer (user\_name, password, count\_downloads, count\_failures). Ο count\_downloads είναι ένας μετρητής ο οποίος αυξάνεται +1 κάθε φορά που ο tracker ενημερώνεται ότι έγινε μία αποστολή αρχείου από τον peer με το συγκεκριμένο user\_name σε έναν άλλο. Ο count\_failures είναι ένας μετρητής ο οποίος αυξάνεται +1 κάθε φορά που ο tracker ενημερώνεται ένα simpleDownload το οποίο είχε ζητηθεί από τον peer με το συγκεκριμένο user\_name απέτυχε.
2. Υποστηρίζει τις λειτουργίες register, login, logout έτσι ώστε να εξυπηρετεί τις αντίστοιχες αιτήσεις των peers.
3. Διατηρεί κατάλληλη δομή δεδομένων που περιέχει τις πληροφορίες επικοινωνίας καθενός peer που είναι συνδεδεμένος αυτή τη στιγμή (token\_id, ip\_address, port, user\_name) καθώς και τους αντίστοιχους μετρητές του count\_downloads και count\_failures.
4. Διατηρεί κατάλληλη δομή δεδομένων με τα ονόματα όλων των διαθέσιμων αρχείων. Η δομή δεδομένων αυτή χρησιμοποιείται για τη λειτουργία reply\_list – βλέπε κατωτέρω.
5. Διατηρεί κατάλληλη δομή δεδομένων που αντιστοιχίζει κάθε διαθέσιμο αρχείο, από τα επιτρεπτά αρχεία τα ονόματα των οποίων περιλαμβάνονται στο ListfileDownload.txt, σε μία λίστα με τα token\_id των ενεργών peers που τα διαθέτουν. Η δομή δεδομένων αυτή χρησιμοποιείται για τη λειτουργία reply\_details (βλέπε κατωτέρω). Στην αρχή είναι κενή και ενημερώνεται προοδευτικά καθώς συνδέονται peers με τον tracker και τον ενημερώνουν μέσω της λειτουργίας inform.
6. Υποστηρίζει πολλαπλά νήματα ταυτόχρονα, δημιουργώντας ξεχωριστό νήμα για κάθε peer τον οποίο εξυπηρετεί.
7. Ενημερώνει κατάλληλα τις δομές δεδομένων και τους μετρητές count\_downloads και count\_failures όταν είναι απαραίτητο μετά από κάθε λειτουργία **notify**. Για τον σκοπό αυτό πρέπει να χρησιμοποιηθούν σωστά κατάλληλοι μηχανισμοί κλειδώματος και κατάλληλες δομές δεδομένων (πχ. ConcurrentHashMap) ώστε να αποφευχθούν race conditions και deadlocks.
8. Υλοποιεί τη λειτουργία **reply\_list** όταν ερωτάται από έναν peer μέσω της λειτουργίας list για το ποια αρχεία είναι διαθέσιμα στο σύστημα.
9. Υλοποιεί τη λειτουργία **reply\_details** όταν ερωτάται από έναν peer για ένα αρχείο μέσω της λειτουργίας details. Χρησιμοποιώντας τη δομή δεδομένων από τη λειτουργία 4 ανωτέρω ευρίσκει τους peers που διαθέτουν το αρχείο. Στη συνέχεια επιβεβαιώνει για καθένα από αυτούς ότι δεν έχει αστοχήσει στέλνοντας τους μηνύματα checkActive. Αν κάποιος peer έχει αστοχήσει ενημερώνει κατάλληλα τις δομές δεδομένων και ακυρώνει το token\_id του. Τέλος απαντά με όλους τους εν λειτουργία peers που διαθέτουν το αρχείο ή με μήνυμα αποτυχίας στην περίπτωση που το αρχείο δεν υπάρχει σε κανέναν peer εν λειτουργία.

**Σημείωση:** Σε όλες τις λειτουργίες πρέπει να εμφανίζονται κατάλληλα μηνύματα επιτυχίας ή αποτυχίας. Για ορισμένα από αυτά πρέπει να παρέχετε screenshots (βλ. Γενικές Πληροφορίες).

## 6 Σημειώσεις

Μπορείτε να υποθέσετε ότι από τη στιγμή που κάνει login ένας peer, τα περιεχόμενα του `shared_directory` του αλλάζουν μόνο όταν κατεβάζει ένα νέο αρχείο. Επίσης μπορείτε να υποθέσετε ότι ένα αρχείο ταυτοποιείται πλήρως από το όνομά του, και επομένως δεν υπάρχουν δύο αρχεία με ίδιο όνομα.

## 7 Υποδοχές (Sockets)

Ο tracker και οι peers θα επικοινωνούν μεταξύ τους μέσω sockets. Μια σύνδεση με sockets είναι ουσιαστικά μία σύνδεση δύο προγραμμάτων μέσω ενός δικτύου. Η υλοποίηση των sockets από την `java.net` σας επιτρέπει να διαχειρίζεστε τα sockets σαν ένα κανάλι ανταλλαγής δεδομένων. Υπάρχουν δύο κλάσεις στο πακέτο `java.net`, η `Socket` και η `ServerSocket`.

Μία τυπική ροή client-server με sockets περιγράφεται ως εξής:

- Ο server ακούει σε ένα `ServerSocket`, το οποίο έχει συσχετιστεί με μία συγκεκριμένη θύρα.
- Ο client χρησιμοποιεί ένα socket για να συνδεθεί στην πόρτα που ακούει ο server. ο client πρέπει ήδη να ξέρει το hostname του υπολογιστή που τρέχει το `ServerSocket` καθώς και το port number του `ServerSocket`.
- Το `ServerSocket` του server αποδέχεται τη σύνδεση του client και δημιουργεί ένα νέο socket με το οποίο μπορεί να επικοινωνεί με τον client. Προσοχή: το `ServerSocket` υπάρχει ακόμα και ακούει για νέους clients.
- Ο client και ο server μπορούν τώρα να επικοινωνούν διαβάζοντας και γράφοντας μέσω των sockets τους.
- Τέλος κλείνουν αμφότεροι τα socket τους.

Στην συγκεκριμένη εφαρμογή:

- Ο tracker θα ακούει σε ένα `ServerSocket`, το οποίο έχει συσχετιστεί με μία συγκεκριμένη θύρα.
- Κάθε peer ακούει και αυτός σε ένα `ServerSocket` στο οποίο περιμένει είτε αιτήσεις κατεβάσματος αρχείου από άλλους peers είτε μηνύματα `checkActive` από τον Tracker ή από άλλους peers.
- Με άλλα λόγια θα υπάρχουν δύο κατηγορίες συνδέσεων ανάλογα με το ποιος εκκινεί τη σύνδεση και με ποιον συνδέεται. Για τις λειτουργίες `register`, `login`, `addFile`, `search` κλπ. τη

σύνδεση την εκκινεί ένας peer και συνδέεται με το ServerSocket του tracker. Για τη λειτουργία κατεβάσματος αρχείου (ή ενός κομματιού αυτού στην Φάση 2 της εργασίας) ο ενδιαφερόμενος peer εκκινεί τις συνδέσεις και συνδέεται με τα ServerSocket των peer που διαθέτουν το αρχείο. Επίσης ο tracker και οι peers μπορούν να εκκινήσουν μια σύνδεση με έναν peer στο ServerSocket του, κάθε φορά που θέλουν να στείλουν μηνύματα checkActive.

- Η μόνη θύρα που πρέπει να θεωρείται εξ' αρχής γνωστή είναι αυτή στην οποία ακούει ο Tracker. Όλες οι υπόλοιπες μπορούν να επιλέγονται τυχαία.

Για περισσότερες πληροφορίες σχετικά με τα sockets μπορείτε να ανατρέξετε στα:

<http://docs.oracle.com/javase/tutorial/networking/sockets/index.html>,

[http://wiki.treck.com/Introduction\\_to\\_BSD\\_Sockets](http://wiki.treck.com/Introduction_to_BSD_Sockets)

### Χρήσιμες τάξεις στη Java

- java.net.ServerSocket
- java.net.Socket
- java.io.DataInputStream
- java.io.DataOutputStream
- java.io.File
- java.io.FileInputStream
- java.io.FileOutputStream
- java.io.ObjectInputStream
- java.io.ObjectOutputStream
- java.util.concurrent.ConcurrentHashMap<K, V>

## 8 Γενικές Πληροφορίες

Θα πρέπει να παραδώσετε σε ηλεκτρονική μορφή όλα τα αρχεία πηγαίου κώδικα (με επαρκή σχολιασμό) της εφαρμογής και σχετικά βοηθητικά αρχεία. Επίσης θα πρέπει να παραδοθεί αναφορά που θα περιέχει:

1. Σύντομη ανάλυση της υλοποίησης.
2. Συνοπτική περιγραφή των αρχείων που αποτελούν την εφαρμογή, οδηγίες για το πως σχετίζονται, πως μεταγλωττίζονται και ποια είναι τα αρχεία εξόδου που προκύπτουν.
3. Τεκμηρίωση της εφαρμογής, π.χ. τρόπος εκτέλεσης προγραμμάτων, προβλήματα που αντιμετωπίσατε, αποκλίσεις από τις προδιαγραφές κτλ.
4. Screenshots της εξόδου που παράγει η εφαρμογή στις ακόλουθες περιπτώσεις:

- Register ενός peer στον tracker (από την πλευρά του peer και του tracker).
  - Login ενός peer στον tracker (από την πλευρά του peer και του tracker).
  - Logout ενός peer στον tracker (από την πλευρά του peer και του tracker).
  - Επιτυχής ερώτηση για τα διαθέσιμα αρχεία και λήψη της σχετικής απάντησης (από την πλευρά του peer και του tracker).
  - Επιτυχής αναζήτηση για ένα αρχείο και λήψη της σχετικής απάντησης (από την πλευρά του peer και του tracker).
  - Ανεπιτυχής αναζήτηση για ένα αρχείο λόγω αστοχίας του μοναδικού peer που είχε το αρχείο (από την πλευρά του peer και του tracker) .
  - Επιτυχές download ενός αρχείου (από την πλευρά του peer που το λαμβάνει και του peer που το στέλνει).
  - Ανεπιτυχές download ενός αρχείου (από την πλευρά του peer που το λαμβάνει και του peer που το στέλνει).
  - Απαραίτητα ενημερωτικά μηνύματα στις διάφορες περιπτώσεις που περιεγράφηκαν ανωτέρω.
5. Γενικότερη αξιολόγηση του αντικειμένου της εργασίας, π.χ. σε ποιους τομείς συναντήσατε κάποια δυσκολία, τι σάς διευκόλυνε στην υλοποίηση, άλλες εναλλακτικές που εξετάσθηκαν αλλά τελικά δεν υλοποιήθηκαν για την ανάπτυξη της εφαρμογής κτλ. .

## 9 Παράδοση – Εξέταση

Η παράδοσης της Φάσης 1 εργασίας θα πρέπει να γίνει μέχρι τις 22/4/2021. Τα παραδοτέα θα υποβληθούν μέσω του eclass ως αρχείο τύπου .zip με όνομα της μορφής <αριθμός\_ομάδας.zip>. **Σύντομα θα ακολουθήσει προφορική εξέταση διάρκειας περίπου 15' ανά ομάδα μέσω τηλεδιάσκεψης.**