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 2)

Ημερομηνία Ανάθεσης: 28/4/2021

Ημερομηνία Παράδοσης: 25/5/2021

1 Περιεχόμενο

Αντικείμενο της παρούσας φάσης της εργασίας είναι η επέκταση των λειτουργιών των peers και του tracker που υλοποιήσατε στην πρώτη φάση της προκειμένου να υποστηρίζουν τον συνεργατικό διαμοιρασμό αρχείων (collaborativeDownload) μέσω ανταλλαγής τμημάτων αυτού (όπως στο Bittorrent), ο οποίος γίνεται ταχύτερα από το "άμεσο" **simpleDownload**, δηλ. την πλήρη λήψη του αρχείου από έναν peer που το διαθέτει ήδη ολόκληρο.

2 Γενικές Πληροφορίες

Η παρούσα φάση της εργασίας θα πρέπει να εκπονηθεί από τις ίδιες ομάδες οι οποίες υλοποίησαν και την πρώτη φάση της. Για απορίες μπορείτε να απευθυνθείτε στο pittaras@aueb.gr. Επίσης, μετά την παράδοση της εργασίας θα ακολουθήσει και προφορική εξέταση της στο εργαστήριο που θα περιλαμβάνει σύντομη περιγραφή της εργασίας, των βημάτων που αναπτύχθηκαν, των δυσκολιών που αντιμετωπίσθηκαν κλπ.

3 Λειτουργίες του Peer

Σε αυτή τη φάση της εργασίας θεωρούμε ότι υπάρχουν τουλάχιστον 10 peers τους οποίους πρέπει να υλοποιήσετε, και 4 διαθέσιμα αρχεία καθένα από τα οποία το έχει κατ' αρχήν μόνο ένας συγκεκριμένος peer ο οποίος και χαρακτηρίζεται ως ο **αρχικός seeder** του αρχείου. (Άρα, η αρχική

κατανομή αρχείων είναι διαφορετική από την πρώτη Φάση της εργασίας.) Καθένα από τα αρχεία αυτά (μεγέθους περί τα 5 MB) θα πρέπει να τεμαχιστεί από τον seeder του σε 10 τμήματα μεγέθους περίπου 0.5 MB. Κάθε τμήμα ταυτοποιείται με βάση το αρχείο στο οποίο ανήκει και έναν αύξοντα αριθμό (π.χ. το 1^{ο} τμήμα του αρχείου keimenoA.txt ονομάζεται keimenoA-1.txt). Ο στόχος είναι τελικά όλοι οι peers να «κατεβάσουν» καθένα από τα αρχεία (δηλαδή όλα τα τμήματα κάθε αρχείου) συνεργαζόμενοι τόσο μεταξύ τους όσο και με τον αρχικό seeder του αρχείου. Η σειρά λήψης των αρχείων αυτών είναι τυχαία, δηλαδή ένας peer ξεκινά να «κατεβάζει» ένα τυχαία επιλεγόμενο αρχείο και μόλις λάβει όλα τα τμήματά του (και με όποια σειρά συμβεί αυτό) και συνενώσει το αρχείο, τότε ξεκινά να «κατεβάζει» ένα άλλο τυχαία επιλεγόμενο αρχείο από τα υπόλοιπα κ.ο.κ. Συστήνεται να χρησιμοποιήσετε μεγάλα αρχεία .txt, ώστε να είναι εύκολος ο τεμαχισμός τους και η επανένωση τους. Για παράδειγμα, μπορείτε να συνενώσετε σε ένα μεγάλο αρχείο .txt πολλαπλά IEFT RFCs.

Συνεπώς, θα πρέπει να επεκτείνετε τις λειτουργίες του peer που έχετε ήδη υλοποιήσει κατά την πρώτη φάση της εργασίας προκειμένου να υποστηρίζει επίσης τις εξής λειτουργίες [**Σημείωση:** Σε όλες τις λειτουργίες πρέπει να εμφανίζονται κατάλληλα μηνύματα επιτυχίας ή αποτυχίας. Για ορισμένα από αυτά πρέπει να παρέχετε screenshots.]

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 τμήματα αρχείων που μοιράζεται με άλλους.
3. Δημιουργεί και διατηρεί ένα λογαριασμό στον tracker (υλοποίηση λειτουργίας **register**, **login** και **logout**), όπως έχουν ήδη υλοποιηθεί.
4. Υλοποιεί τη λειτουργία **list**, όπως έχει ήδη υλοποιηθεί, με την οποία ζητά από τον tracker την λίστα με τα ονόματα των αρχείων που είναι διαθέσιμα συνολικά στο σύστημα P2P.
5. 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 την εξής εμπλουτισμένη σε σχέση με την πρώτη φάση λίστα:
    a. μία λίστα με τις πληροφορίες επικοινωνίας (ip_address, port, user_name, count_downloads, count_failures, {pieces}, seeder-bit) καθενός από τους peers που έχουν τμήματα (pieces) του συγκεκριμένου αρχείου, όπου {pieces} υποδηλώνει το σύνολο των τμημάτων που κατέχει ο εκάστοτε peer, και το seeder-bit αποτελεί ένδειξη αν ο peer αυτός είναι ο αρχικός seeder του αρχείου[1],
    b. μήνυμα αποτυχίας στην περίπτωση που το αρχείο δεν υπάρχει σε κανέναν άλλο peer.
6. Υλοποιεί τη λειτουργία **checkActive και** απαντά με μήνυμα επιβεβαίωσης στα μηνύματα checkActive που λαμβάνει**,** όπως έχει ήδη υλοποιηθεί.

[1] Το seeder-bit ενός peer ορίζεται για καθένα αρχείο, με τιμή 0 ή 1, η οποία δεν αλλάζει. Το {pieces} ενός peer ορίζεται για καθένα αρχείο, και αν ο peer δεν είναι αρχικός seeder του αρχείου αυτού τότε το σύνολο αυτό "διευρύνεται¨ σταδιακά.

7. Εάν είναι ο αρχικός seeder του αρχείου υλοποιεί την νέα λειτουργία **partition**, δηλαδή τεμαχίζει το αρχείο σε τμήματα μεγέθους περίπου 1 MB το καθένα.
8. Εάν είναι αρχικός seeder για ένα (ή περισσότερα) αρχεία υλοποιεί την νέα λειτουργία **seeder-inform1:** Όταν seeder λάβει το token_id από τον tracker μετά από το πρώτο επιτυχές **login** ενημερώνει τον tracker για τις πληροφορίες επικοινωνίας (ip_address, port) για τα τρέχοντα περιεχόμενα του shared_directory του (δηλ. ποιων αρχείων κατέχει όλα τα τμήματα καθώς και τα ονόματα των τμημάτων αυτών) και για το ότι μπορεί να λειτουργήσει ως seeder για τα αρχεία αυτά.
9. Εάν είναι seeder (όχι απαραίτητα ο αρχικός) για ένα αρχείο (ή περισσότερα) υλοποιεί την νέα λειτουργία **seeder-serve**: Ένας seeder μπορεί επίσης να δεχθεί αιτήματα από τους υπόλοιπους peers για αποστολή τμημάτων του αρχείου του οποίου αποτελεί seeder. Μόλις δεχθεί ένα τέτοιο αίτημα, αναμένει για 200msec μήπως λάβει επιπλέον αιτήματα. Μετά το πέρας του χρόνου αυτού, ο seeder επιλέγει τυχαία σε ποιον peer θα στείλει ένα μόνο από τα ζητούμενα τμήματα αρχείου. Η επιλογή του τμήματος του αρχείου που θα στείλει γίνεται τυχαία ανάμεσα στα τμήματα που τού ζητώνται στα πλαίσια ενός αιτήματος.
10. Υλοποιεί τη νέα λειτουργία **select**, όπου επιλέγει τυχαία το επόμενο αρχείο το οποίο θα προσπαθεί να «κατεβάσει» (ανάμεσα σε αυτά που δεν έχει ήδη) σε συνεργασία με τους άλλους peers.
11. Υλοποιεί τη νέα λειτουργία **collaborativeDownload.** Ένας peer, όταν δεν είναι seeder ενός αρχείου, δέχεται αίτημα ή αιτήματα από άλλους peers για την αποστολή κάποιου τμήματος του αρχείου αυτού. Πριν ξεκινήσει την αποστολή περιμένει για 200msec μήπως λάβει επιπλέον αιτήματα. Μετά το πέρας του χρόνου αυτού ενεργεί ως εξής:

    Α) Εάν έχει λάβει μόνο ένα αίτημα, αποστέλλει ένα μόνο τμήμα, τυχαία επιλεγόμενο από τα ζητούμενα τμήματα αρχείου που διαθέτει στο shared_directory του, και ζητά από τον peer στον οποίο το αποστέλλει κάποια άλλα τμήματα του ιδίου αρχείου (εάν του λείπει κάποιο), στέλνοντας του κατάλληλο αίτημα (το οποίο θα αντιμετωπισθεί με τον ίδιο τρόπο, στα πλαίσια της λειτουργίας **collaborativeDownload**).

    Β) Εάν έχει λάβει περισσότερα από ένα αιτήματα, τότε αποφασίζει ποιον να εξυπηρετήσει από τους peers που έχουν υποβάλει αίτημα (στέλνοντας πάντα ένα μόνο τμήμα, τυχαία επιλεγόμενο από τα ζητούμενα τμήματα αρχείου που διαθέτει στο shared_directory του), κάνοντας την παρακάτω τυχαία επιλογή:

    - με πιθανότητα p=0.2, επιλέγει τυχαία έναν από τους peers που έχουν υποβάλει αίτημα, και ζητά από τον peer στον οποίο αποστέλλει κάποια άλλα τμήματα του ιδίου αρχείου (εάν του λείπει κάποιο), στέλνοντας του κατάλληλο αίτημα (το οποίο θα αντιμετωπισθεί στα πλαίσια της λειτουργίας **collaborativeDownload**).
    - με πιθανότητα p=0.4, επιλέγει τον καλύτερο peer από αυτούς που έχουν υποβάλει αίτημα, ο οποίος επιλέγεται βάσει του κριτηρίου που εφαρμοζόταν ως τώρα στην λειτουργία **simpleDownload**, και ζητά από τον peer στον οποίο αποστέλλει κάποια άλλα τμήματα του ιδίου αρχείου (εάν του λείπει κάποιο), στέλνοντας του κατάλληλο αίτημα (το οποίο θα αντιμετωπισθεί στα πλαίσια της λειτουργίας **collaborativeDownload**).

- με πιθανότητα p=0.4 επιλέγει εκείνον τον peer από τον οποίο έχει λάβει ως τώρα τα περισσότερα τμήματα αρχείων (διατηρώντας τοπικά κατάλληλη δομή δεδομένων), και ζητά από τον peer στον οποίο αποστέλλει κάποια άλλα τμήματα του ιδίου αρχείου (εάν του λείπει κάποιο), στέλνοντας του κατάλληλο αίτημα στέλνοντας του κατάλληλο αίτημα (το οποίο θα αντιμετωπισθεί, στα πλαίσια της λειτουργίας **collaborativeDownload**). Σε περίπτωση «ισοπαλίας» επιλέγεται ο καλύτερος από τους ισοβαθμούντες peers (βάσει του κριτηρίου που εφαρμοζόταν ως τώρα στην λειτουργία **simpleDownload)**. Για το σκοπό αυτό, κάθε peer διατηρεί μία δομή δεδομένων με πληροφορίες για την λήψη τμημάτων αρχείων από άλλους peers. Συγκεκριμένα διατηρεί πληροφορίες σχετικά με το όνομα κάθε τμήματος αρχείου που έχει λάβει έως τώρα και την ταυτότητα (username) του peer από τον οποίο έχει λάβει το συγκεκριμένο τμήμα.

  Σε καθεμία από τις ανωτέρω περιπτώσεις, αφού κάνει την επιλογή του αιτήματος που θα εξυπηρετήσει, αν δεν διαθέτει κανένα από τα αιτούμενα τμήματα του αρχείου (πχ. γιατί απλώς το ερώτημα του υπεβλήθη εκ παραδρομής) απαντά αρνητικά. Επίσης, απαντά αρνητικά και στους peers που έχουν υποβάλει αίτημα αλλά δεν επιλέγονται προς εξυπηρέτηση.

  Σημειωτέον ότι οι ανωτέρω κανόνες επιλογής λειτουργούν σαν μηχανισμοί επιβράβευσης των peers με την καλύτερη επίδοση και την πιο σημαντική συνεισφορά στο συνεργατική ανταλλαγή των αρχείων.

  Επιπλέον των παραπάνω αιτημάτων που στέλνει ένας peer σαν «αντάλλαγμα» αυτών που εξυπηρετεί, στα πλαίσια της λειτουργίας **collaborativeDownload** κάθε peer μπορεί να στέλνει νέο αίτημα κάθε 500 msec για αποστολή τμημάτων (του αρχείου που «κατεβάζει» τώρα) σε 4 άλλους peers από εκείνους που διαθέτουν τμήματα που τού λείπουν, σύμφωνα με την ενημέρωση που έχει λάβει από τον tracker για διαθεσιμότητα τμημάτων του αρχείου αυτού. Η επιλογή των peers στους οποίους αποστέλλονται τα αιτήματα γίνεται με τον εξής τρόπο: κάθε peer επιλεγεί να στείλει αίτημα στους 2 peers (ανάμεσα σε εκείνους που διαθέτουν ορισμένα τα τμήματα του αρχείου που επιθυμεί να «κατεβάσει», αλλά εκτός του αρχικού seeder του αρχείου) από τους οποίους που έχει λάβει ως τώρα τα περισσότερα τμήματα αρχείων σύμφωνα με τις πληροφορίες της αντίστοιχης δομής δεδομένων (βλέπε ανωτέρω) και σε άλλους 2 peers τυχαία επιλεγόμενους (περιλαμβανομένου και του αρχικού seeder). Εάν οι peers που διαθέτουν τα τμήματα είναι 4 ή λιγότεροι τότε στέλνει αίτημα σε όλους. Εάν δεν εξυπηρετηθεί για κανένα από τα αιτήματα που έχει στείλει, τότε μετά από 500 msec από την λήψη της τελευταίας από όλες τις αρνητικές απαντήσεις (ή μετά από ένα timeout των 2 sec από την αποστολή του αρχικού αιτήματος, εάν εν τω μεταξύ δεν έχει λάβει αρνητικές απαντήσεις για όλα τα αιτήματα), ζητά και λαμβάνει νέα ενημέρωση από τον tracker σχετικά με τα διαθέσιμα τμήματα αρχείων καθώς και από ποιους peers είναι διαθέσιμα και στέλνει νέο αίτημα σε 4 peers σύμφωνα με τον κανόνα που περιγράφηκε παραπάνω.

12. Μετά την επιτυχή λήψη ενός τμήματος αρχείου, ο **peer** το αποθηκεύει στο shared_directory και ενημερώνει τον tracker (τροποποιημένη λειτουργία **notify**) ότι διαθέτει και αυτό το τμήμα αρχείου καθώς και για το user_name του peer από τον οποίο δόθηκε επιτυχώς το τμήμα. Σε

περίπτωση που η λήψη αυτή αποτύχει, ο peer ενημερώνει (λειτουργία **notify**) τον tracker για την αποτυχία, και προχωρά αργότερα στην αποστολή νέου αιτήματος.

13. Μόλις ολοκληρωθεί η λήψη όλων των τμημάτων εντός αρχείου από έναν peer, τότε ο peer αυτός λειτουργεί ως seeder για το αρχείο αυτό, απαντώντας στα σχετικά αιτήματα σύμφωνα με την λειτουργία **seeder-serve** ανωτέρω. Επίσης, ο peer ταξινομεί τα τμήματα, με τη σωστή σειρά, προκειμένου και τα «ενώνει» (νέα λειτουργία **assemble**) ώστε να είναι διαθέσιμα για «προβολή» και επεξεργασία. (Πρέπει να φροντίσετε βάσει της διαθέσιμης πληροφορίας να υλοποιήσετε έναν τρόπο για να διαπιστώνει ένας peer ότι για ένα αρχείο έχει κατεβάσει όλα τα τμήματα.) Τέλος, εκτελεί τη λειτουργία **select.**

4 Λειτουργίες του Tracker

Ο tracker εκτελεί τις εξής λειτουργίες, που γενικά δεν διαφέρουν σημαντικά από αυτές που έχουν ήδη υλοποιηθεί:

1. Διατηρεί κατάλληλη δομή δεδομένων με τις πληροφορίες κάθε εγγεγραμμένου peer (user_name, password, count_downloads, count_failures, {pieces}, seeder-bit). Ο count_downloads είναι ένας μετρητής ο οποίος αυξάνεται +1 κάθε φορά που ο tracker ενημερώνεται ότι έγινε μία αποστολή τμήματος 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 για ότι μια αποστολή τμήματος αρχείου από τον peer με το συγκεκριμένο user_name απέτυχε. Επίσης το {pieces} υποδηλώνει το σύνολο των τμημάτων που κατέχει ο εκάστοτε peer, και το seeder-bit αποτελεί ένδειξη αν ο peer αυτός είναι ο αρχικός seeder του αρχείου.
2. Υποστηρίζει τις λειτουργίες register, login, logout έτσι ώστε να εξυπηρετεί τις αντίστοιχες αιτήσεις των peers, όπως έχουν ήδη υλοποιηθεί.
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, το σύνολο {pieces}, και το seeder-bit.
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 διαθέτουν τμήματα του αρχείου. 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 seeder-inform1 (αρχικά οι seeders) και 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 1 ανωτέρω ευρίσκει τους peers που διαθέτουν τμήματα του αρχείου. 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 τμήματα του αρχείου ή με μήνυμα αποτυχίας στην περίπτωση που τμήματα του αρχείου δεν υπάρχουν 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).

5 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 τμήματος αυτού στην Φάση 2 της εργασίας) ο ενδιαφερόμενος peer εκκινεί τις συνδέσεις και συνδέεται με τα ServerSocket τών 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

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>

6 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 τμήματα ενός αρχείου και λήψη της σχετικής απάντησης (από την πλευρά του peer και του tracker).
   - Ανεπιτυχής αναζήτηση (από την πλευρά του peer και του tracker) για τμήματα ενός αρχείου λόγω αστοχίας των peers που τα κατέχουν και του αρχικού seeder
   - Επιτυχές download ενός τμήματος αρχείου (από την πλευρά του peer που το λαμβάνει και του peer που το στέλνει).
   - Ολοκλήρωση του πλήρους download ενός αρχείου (δηλαδή όλων των τμημάτων του).
   - Ανεπιτυχές download ενός τμήματος 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. .

7 Παράδοση – Εξέταση

Η παράδοσης της Φάσης 2 της εργασίας θα πρέπει να γίνει μέχρι τις 25/5/2021. 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.**