

## Αναφορά 2ης φάσης προγραμματιστικής εργασίας

**Φίλιππος Δουραχαλής 3170045**

**Σπυριδούλα Κουτσούκου 3170080**

Το πρόγραμμα υλοποιήθηκε σε περιβάλλον IntelliJ IDEA. Πριν την εκτέλεση του κώδικα χρειάζεται να φορτώσουμε τις βιβλιοθήκες jar commons-io και mp3agic που δίνονται εντός του παραδοτέου.

Για την εκπόνηση της εργασίας έχουμε υποθέσει ότι :

1. Αν βρεθεί ολόκληρο αρχείο στο SharedResources τότε αυτός είναι ο αρχικός seeder.
2. Τα αρχεία assembled μέσω της μεθόδου assemble τοποθετούνται στο φάκελο Downloads του peer ώστε να μην μπερδεύονται με τα αρχεία για τα οποία αποτελεί αρχικό seeder.
3. Οι αρχικοί seeders των 4 αρχείων είναι οι: jojo, alice, bob και phil  
Οι κωδικοί τους αναφέρονται στο output του tracker και το αρχείο accounts.txt (δεύτερο πεδίο) για εύκολη πρόσβαση.

```
jojo, 12345, 26, 3860
phil, 991510, 94, 212
alice, aes256, 143, 3702
bob, s21kub, 2, 3884
danny, d4c, 0, 307
bill, bill, 0, 280
maria, netrulz, 0, 261
sam, pineapplepizza, 0, 261
mista, 6pistols, 0, 261
Risotto Nero, metallica, 0, 209
RisottoNero, metallica, 0, 209
Waiting for the client request
Awaiting client connection...
Start
Awaiting client connection...
Start
Awaiting client connection...
Client
login
Username: phil
received password: 991510
991510
ip: 192.168.1.165
port: 3924
Client
login
Username: alice
received password: aes256
aes256
ip: 192.168.1.165
port: 6162
```

Όπως και στην προηγούμενη φάση της εργασίας, τρέχουμε αρχικά τον tracker ο οποίος αναμένει συνδέσεις από peers. Κάθε peer που συνδέεται έχει την επιλογή να συνδεθεί (login) ή να εγγραφεί (register) αν δεν είναι ήδη στο σύστημα. Κάθε νέος χρήστης τοποθετείται σαν ξεχωριστή εγγραφή στο αρχείο accounts.txt ώστε ο tracker να «θυμάται» όλες τις πληροφορίες (username, password, count\_downloads, count\_failures) ενός χρήστη ακόμα και μετά τον τερματισμό του προγράμματος.

Σε κάθε περίπτωση, όταν ένας χρήστης συνδεθεί του αποδίδεται κανονικά το token\_id όπως στην φάση Α' και εκείνος με τη σειρά του στέλνει τα περιεχόμενα του sharedResources, τα οποία είναι άμεσα διαθέσιμα στους συνδεδεμένους peers.

Κατά το logout ο tracker ενημερώνει τις δομές του σχετικά με τον peer που αποσυνδέθηκε ώστε να παρουσιάζει μια συνεπή κατάσταση του συστήματος στους υπόλοιπους peers.

```
Logout? [Y/N]
y
Attempting Connection...
Connected to:192.168.1.165 2300
Got I/O streams
Successful logout
```

Στην κλάση Peer έχουμε κάνει τις παρακάτω τροποποιήσεις:

## Κλάση Peer

### Δομές δεδομένων:

1. *Map<String, ChunksWrapper> directoryListing*  
Πρόκειται για την σημαντικότερη δομή του peer, όπου αποθηκεύονται για κάθε αρχείο (String) τα chunks από τα οποία αποτελείται καθώς και αν για το αρχείο αυτό είναι αρχικός seeder. Οι δύο αυτές πληροφορίες περικλείονται σε μια βοηθητική κλάση ChunksWrapper
2. *Map<String, Integer> filechunks*  
Αποθηκεύει τον αριθμό των chunks ενός αρχείου, όπως ενημερώνεται από τον tracker και όπως τον υπολογίζει ο ίδιος (αν για το αρχείο είναι αρχικός seeder)
3. *List<PeerDetails> servingClients*  
Οι άλλοι peers τους οποίους εξυπηρετεί τη δεδομένη χρονική στιγμή ο peer. Χρησιμοποιείται για να επιλεγθεί ένας μόνο που θα εξυπηρετηθεί τελικά.
4. *Map<String, List<Integer>>*  
Αποθηκεύει για τον peer, από ποιους άλλους peers έχει λάβει ποια τμήματα αρχείων (List<Integer>).

### Μέθοδοι:

1. *searchDirectory(File directory)*  
Επεκτείνεται η λειτουργικότητα της μεθόδου ώστε ο peer να είναι σε θέση να εντοπίσει τα chunks του αρχείου και να τα προσθέσει στο directoryListing του. Επίσης μπορεί να καταλάβει αν για το αρχείο αυτό αποτελεί ο αρχικός seeder.
2. *selectPeer()*  
Διαβάζει τη λίστα servingClients και αποφασίζει ποιος peer θα εξυπηρετηθεί, όπως ορίζεται στην εκφώνηση.
3. *collaborationDownload(String filename)*  
Είναι υπεύθυνη για το κατέβασμα του αρχείου που έχει επιλεγθεί. Αρχικά επιλέγει 4 peers ως ορίζεται στην εκφώνηση και στην συνέχεια στέλνει σε όλους αιτήματα για τα chunks του αρχείου που του λείπουν. Επίσης θέτει ένα timer 2 δευτερολέπτων ώστε σε περίπτωση που δεν απαντήσει ο server με τον οποίο επικοινωνεί να προκύψει timeout. Σε αντίθετη περίπτωση κατεβάζει το chunk και το αποθηκεύει στο shared resources φακέλο του.

4. *select()*  
Επιλέγει ένα τυχαίο αρχείο για το οποίο ο peer δεν είναι ήδη seeder για να το κατεβάσει
5. *partition(String filename)*  
Τεμαχίζει το αρχείο που δόθηκε ως είσοδος, εφόσον αποτελεί seeder για αυτό (και άρα το έχει ολόκληρο)

### Κλάση PeerDetails

Η κλάση PeerDetails πλέον περιέχει όλα τα αρχεία, μαζί με τα chunks που έχει στην κατοχή του ένας peer (ένα Map δηλαδή αντίστοιχο του directoryListing). Με αυτόν τον τρόπο ο tracker μπορεί να ενημερώνει αποτελεσματικότερα τους peers για το ποιος είναι αρχικός seeder και ποια αρχεία υπάρχουν που χωρίς να επιβαρυνόμαστε με επιπλέον δομές.

### Κλάση ChunkFile

Αντιπροσωπεύει ένα τμήμα του αρχείου ως byte array. Χρησιμοποιείται για την αποθήκευση και το διάβασμα. Η υποκλάση της **MusicFile** αντιπροσωπεύει συγκεκριμένα ένα μουσικό αρχείο mp3 που απαιτεί επιπλέον πληροφορίες για να αποθηκευτεί.

Ο Peer σηκώνει επίσης ένα νέο non-blocking νήμα για κάθε σύνδεση που δέχεται. Μόλις λάβει αίτημα για download, ζητάει από τον tracker να τον ενημερώσει με ποιόν peer επικοινωνεί και στην συνέχεια τον προσθέτει στη λίστα με τους εξυπηρετούμενους peers. Με αυτόν τον τρόπο μπορεί να επιλέξει ανάμεσά τους έναν τυχαίο. Αν ένας peer δεν επιλεγθεί, ο server απλά επιστρέφει και από την πλευρά του μη-επιλεγθέντος peer συμβαίνει timeout. Σημειώνουμε σε αυτό το σημείο ότι ενώ η download δουλεύει για μεμονωμένα αιτήματα, μας ήταν δύσκολο να πραγματοποιήσουμε πολλά ταυτόχρονα αιτήματα για το ίδιο αρχείο για να επαληθεύσουμε ότι δουλεύει 100% σωστά. Επίσης δεν έχει υλοποιηθεί η λειτουργία ο server-peer να ζητάει «αντάλλαγμα» όταν δεν είναι seeder του αρχείου που του ζητείται.

### Ας περάσουμε τώρα στην εκτέλεση του προγράμματος.

Το πρόγραμμα δύναται να δουλέψει με αρχεία .mp3 και .txt, ωστόσο στην περίπτωση των mp3 δεν είναι δυνατή η αναπαραγωγή των ληφθέντων αρχείων.

1. Στη main() της κλάσης Peer αφού ο peer καλεί την searchDirectory() για να διαβάσει τα αρχεία και να καταλάβει για ποια είναι seeder ,μέσα στην searchDirectory() καλείται η partition η οποία τεμαχίζει το αρχείο σε πακέτα.

```
SharedResources contains:
Stereotype News.mp3 (1)
there are 2629082 bytes in file
TOTAL BYTES = 2629082
Splitting into 5 chunks
0
512000
1024000
1536000
2048000
2117082
```

Τα login, register και logout γίνονται κανονικά όπως στην φάση Α' της εργασίας

```
Attempting Connection...
Connected to:192.168.1.117 2300
Got I/O streams
Do you want to register or login
register
Please choose a username:
danny
Please select a password
d4c
Register successful!
There are not any files to share yet!
Do you want to download a file? [Y/N]
Waiting for clients to connect...
y
Files currently available in the P2P System:
1) Stereotype News.mp3
2) Pickled Pink.mp3
```

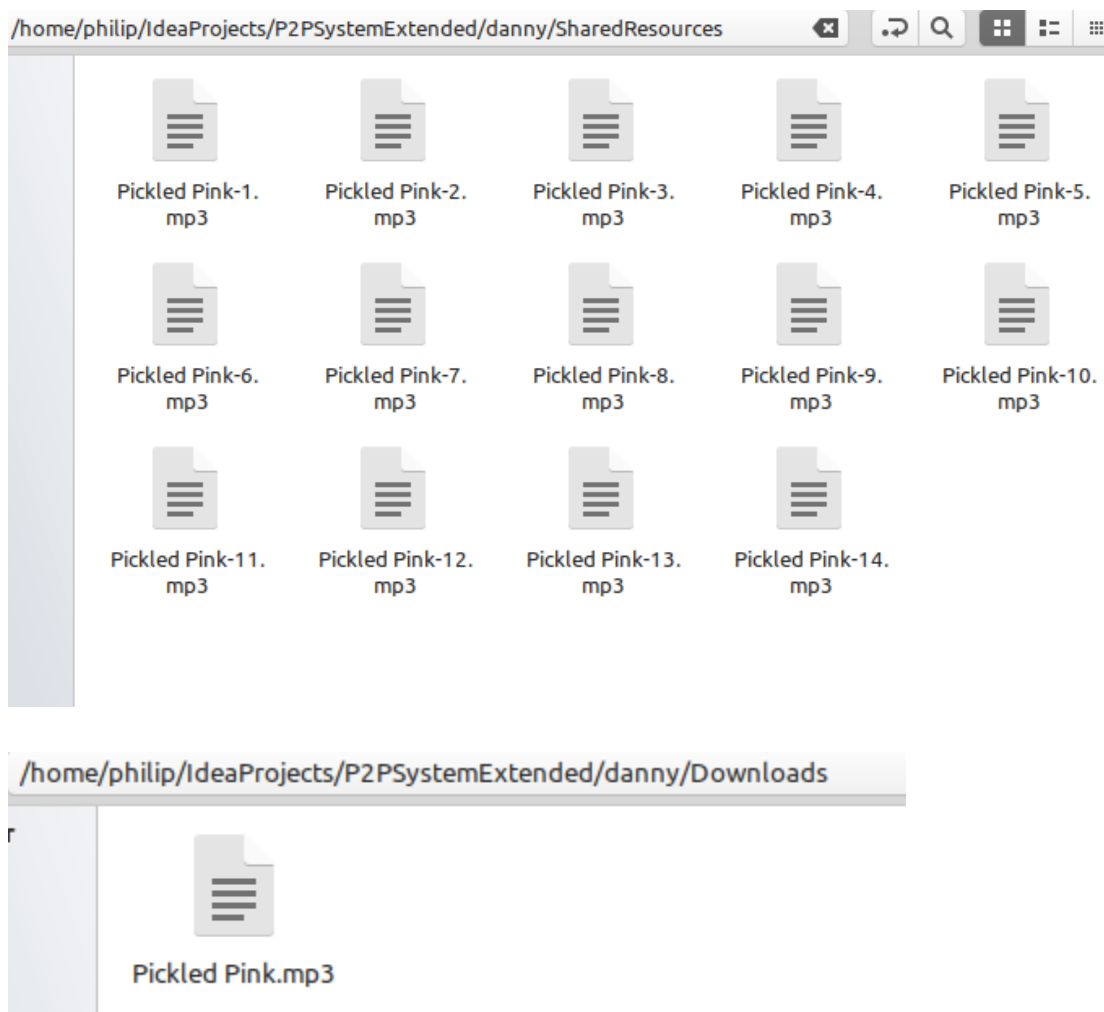
```
Attempting Connection...
Connected to:192.168.1.117 2300
Got I/O streams
Do you want to register or login
login
Please enter your username:
alice
Please enter your password
aes256
You are now logged in.
Your id for this session is: 391
```

2. Επικοινωνεί με τον tracker για να πάρει την λίστα με τα αρχεία και μέσω της select επιλέγει ένα τυχαίο αρχείο για το οποίο δεν είναι seeder.

```
SELECTED FILE = Pickled Pink.mp3
Pickled Pink.mp3
Downloading chunk
Requesting file
File exists: true
These peers have the file:
```

```
Username: alice
IP: 192.168.1.117
Port: 4945
Successful Downloads: 14
Failed Downloads: 3547
```

3. Έπειτα συνδέεται με το πολύ 4 peer από αυτούς που του έχει επιστρέψει ο tracker και καλεί την collaborativeDownload μέχρι να κατεβάσει όλο το αρχείο και η asseble για να γίνει η επανένωσή και να αποθηκευτεί . Όπως αναφέραμε, το αρχείο προορισμού της επανένωσης βρίσκεται στο φάκελο «Downloads», ενώ τα chunks αποθηκεύονται κανονικά στον SharedResources του peer.



4. Ξανά καλείται η select εάν υπάρχουν και αλλά αρχεία για τα οποία δεν είναι seeder ο peer και επαναλαμβάνεται η ίδια διαδικασία για το κατέβασμα όλων των πακέτων. Σημειώνεται ότι ο χρήστης μπορεί να επιλέξει μόνο αν θέλει να κατέβει κάποιο επιπλέον αρχείο, όχι ποιο θα είναι αυτό (καθώς η επιλογή γίνεται τυχαία από την select).

Pickled Pink-1.mp3  
Pickled Pink-10.mp3  
Pickled Pink-11.mp3  
Pickled Pink-12.mp3  
Pickled Pink-13.mp3  
Pickled Pink-14.mp3  
Pickled Pink-2.mp3  
Pickled Pink-3.mp3  
Pickled Pink-4.mp3  
Pickled Pink-5.mp3  
Pickled Pink-6.mp3  
Pickled Pink-7.mp3  
Pickled Pink-8.mp3  
Pickled Pink-9.mp3

```
Number of chunks for this file: 6
Stereotype News-1.mp3
ADDINGStereotype News-1.mp3
Stereotype News-2.mp3
Stereotype News-3.mp3
Stereotype News-4.mp3
ADDINGStereotype News-4.mp3
Stereotype News-5.mp3
ADDINGStereotype News-5.mp3
Stereotype News-6.mp3
choosing random peer
ADDING PEER:
Username: phil
IP: 192.168.1.165
Port: 3924
Successful Downloads: 97
Failed Downloads: 212
```

```
Username: phil
IP: 192.168.1.165
Port: 3924
Successful Downloads: 97
Failed Downloads: 212
```

```
Downloading file, please wait...
Attempting Connection...
Connected to:192.168.1.165 3924
Got I/O streams
*****Download complete!*****
```

*κατέβασμα ενός chunk*

5. Όταν ο server λάβει μια σύνδεση, επιλέγει ανάμεσα στους peers έναν μόνο και ανάμεσα στα chunks που ζητάει ο peer ένα μόνο για να του στείλει (εφόσον το έχει). Αν είναι seeder για το αρχείο του chunk, δεν πραγματοποιεί κάποια άλλη ενέργεια.



