

Κωδικός Ομάδας 1006

Στα πλαίσια του project υλοποιήσαμε μια εφαρμογή διαμοιρασμού αρχείων μεταξύ διαφορετικών υπολογιστών(p2p file sharing). Για την χρήση της εφαρμογής ο κάθε χρήστης χρειάζεται να δημιουργήσει έναν λογαριασμό και στην συνέχεια να συνδεθεί με αυτόν. Κάθε λογαριασμός αντιστοιχεί σε ένα στιγμιότυπο της κλάσης Peer. Οι Peers επικοινωνούν με τον Tracker ο οποίος κρατάει πόσοι Peers είναι γραμμένοι και πόσοι συνδεδεμένοι κάθε στιγμή. Απαγορεύεται να έχουν 2 peers ίδιο όνομα. Για την επιτήρηση των Peer ο Tracker αποθηκεύει σε δική του δομή στιγμιότυπα της κλάσης Peer_info που αναπαριστά τους Peer με τις ελάχιστες απαραίτητες πληροφορίες για την εγκαθίδρυση σύνδεσης σε αυτούς και την διατήρηση των λογαριασμών τους. Προκειμένου κάποιος Peer να εκτελέσει κάποια λειτουργία στέλνει αντίστοιχο μήνυμα με την λειτουργία και τα απαραίτητα στοιχεία που χρειάζονται για την πραγματοποίηση της στον Tracker σαν plain text. Ο Tracker στην συνέχεια επικοινωνεί αν αυτό είναι απαραίτητο με τον αντίστοιχο Peer (πχ. στην περίπτωση απαρύθμισης των αρχείων που είναι διαθέσιμα για Ενναλακτικά 2 peer μπορούν να επικοινωνούν μεταξύ τους αφού κάθε peer επιτελεί και λειτουργίες server για την απόκριση σε μηνύματα CheckActive ή αποστολή αρχείων. Κατά την δημιουργία του project δημιουργήθηκαν 3 αρχεία Peer, Tracker και Peer_info. Ακόμα υπάρχει ένας φάκελος στον οποίο περιέχονται φάκελοι που αντιστοιχούν ο καθένας σε έναν από τους Peers που χρησιμοποιούν την εφαρμογή. Μέσα στον κάθε φάκελο του κάθε Peer βρίσκονται τα αρχεία τα οποία έχει ο συγκεκριμένος κατεβασμένα και έτοιμα για διαμοιρασμό. Κατά το download ενός αρχείου αυτό δημιουργείται στον φάκελο του αντίστοιχου Peer που το κατέβασε. Κάθε peer έχει δικό του directory peer_i όπου i είναι ένα constructor argument στο Peer αυτό το κάναμε για ευκολία στην εκτέλεση του προγράμματος τοπικά αλλά δεν θα χρειαζόταν όταν στήναμε πραγματικά το σύστημα. Όπου χρειάζεται επικοινωνία μεταξύ Peer και Tracker ή Peer και Peer αυτή επιτυγχάνεται μέσω socket της Python. Κάθε φορά που συνδέεται ένας Peer στην εφαρμογή, αυτόματος συνδέεται και με τον Tracker με του server socket του δεύτερου. Ακόμα κάθε φορά που 2 Peers χρειάζεται να συνδεθούν μεταξύ τους εκείνος που θέλει να κατεβάσει το αρχείο συνδέεται στο server socket αυτού που το έχει διαθέσιμο.

SCREENSHOTS

REGISTER & LOGIN

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\AUEB\σ_εξάμηνο\diktya> & 'c:\Users\THODORIS\AppData\Local\Programs\Python\Python38\python.exe' ..\..\debugpy\launcher '53814' '--' 'd:\AUEB\σ_εξάμηνο\diktya\Peer.py'
WELCOME TO THE PEER-TO-PEER FILE SHARING SYSTEM !
Enter 'R' for registration or 'L' for login: r

Registration:
Please enter your name: 123
Please enter your password: 123
Registration successful
Assigned token_id: 335
['file1.txt']

Peer informed the tracker about its shared files successfully.

Peer Menu:
1. List available files
2. Download a file
3. File details
4. Logout
5. Exit

Enter your choice (1-5): █
```

```
PS D:\AUEB\σ_εξάμηνο\diktya> & 'c:\Users\THODORIS\AppData\Local\Programs\Python\Python38\python.exe' ..\..\debugpy\launcher '53791' '--' 'd:\AUEB\σ_εξάμηνο\diktya\Tracker.py'
Tracker server listening on 127.0.0.1:65432
New connection from ('127.0.0.1', 53818)
New connection from ('127.0.0.1', 53819)
connected :
SUCCESS 335
New connection from ('127.0.0.1', 53820)
{'file1.txt': {335}}
```

LOGOUT

```
Enter your choice (1-5): 4

Logging out...
New connection from ('127.0.0.1', 65293)
OK

RESULT SUCCESSFUL LOGOUT
New connection from ('127.0.0.1', 65295)
OK

Logged out successfully.
WELCOME TO THE PEER-TO-PEER FILE SHARING SYSTEM !
Enter 'R' for registration or 'L' for login: █
```

ο Tracker τυπώνει τις ανανεωμένες δομές διαθέσιμων αρχείων

```
New connection from ('127.0.0.1', 65245)
[]
New connection from ('127.0.0.1', 65292)
{'file6.txt': {899}, 'file7.txt': {899}, 'file8.txt': {899}}
{}
New connection from ('127.0.0.1', 65294)
{}
```

AVAILABLE FILES

```
Peer Menu:
1. List available files
2. Download a file
3. File details
4. Logout
5. Exit

Enter your choice (1-5): 1
Available files:
file1.txt
(downloaded
file)
file2.txt

Peer Menu:
1. List available files
2. Download a file
3. File details
4. Logout
5. Exit

Enter your choice (1-5): █
```

```
New connection from ('127.0.0.1', 53837)
New connection from ('127.0.0.1', 53838)
connected :
SUCCESS 414
New connection from ('127.0.0.1', 53839)
{'file1.txt': {335}, 'file2.txt': {414}}
New connection from ('127.0.0.1', 53840)
```

SUCCESSFUL FILE SEARCH(details with 2 peers)

```

Peer Menu:
1. List available files
2. Download a file
3. File details
4. Logout
5. Exit

Enter your choice (1-5): 3

Enter the filename to get peers that host it: file2.txt
RESPONSE: [{username:321,password:321,count_downloads:0,count_failures:0,token_id:414,ip:127.0.0.1,port:9992}]
PeerINFO(username='321', password='321', token_id=414, ip='127.0.0.1', port='9992', count_downloads=0, count_failures=0)
P TIME TO RESPOND:0.43160002678632736 ms
[0.0]

Peer Menu:
1. List available files
2. Download a file
3. File details
4. Logout
5. Exit

Enter your choice (1-5): █

```

```

New connection from ('127.0.0.1', 53837)
New connection from ('127.0.0.1', 53838)
connected :
SUCCESS 414
New connection from ('127.0.0.1', 53839)
{'file1.txt': {335}, 'file2.txt': {414}}
New connection from ('127.0.0.1', 53840)
New connection from ('127.0.0.1', 54105)
{username:321,password:321,count_downloads:0,count_failures:0,token_id:414,ip:127.0.0.1,port:9992}
[{'username:321,password:321,count_downloads:0,count_failures:0,token_id:414,ip:127.0.0.1,port:9992}]

```

Unsuccessful details/file search

```

Enter your choice (1-5): 1
Available files:
file6.txt
(downloaded
file)
file7.txt
(downloaded
file)
file8.txt
(downloaded
file)

Peer Menu:
1. List available files
2. Download a file
3. File details
4. Logout
5. Exit

Enter your choice (1-5): 2

Enter the filename to download: hasgj
Error retrieving the details of the file: FILE DOES NOT EXIST

```

```

Connected .
SUCCESS 899
New connection from ('127.0.0.1', 65224)
{'file6.txt': {899}, 'file7.txt': {899}, 'file8.txt': {899}}
New connection from ('127.0.0.1', 65225)
New connection from ('127.0.0.1', 65244)
New connection from ('127.0.0.1', 65245)
[]

```

S
U
C
Sender:

```

5. Exit

Enter your choice (1-5): New connection from ('127.0.0.1', 49209)
FILE SENT

```

DOWNLOAD
Receiver:

```

1. List available files
2. Download a file
3. File details
4. Logout
5. Exit

Enter your choice (1-5): 2

Enter the filename to download: file6.txt
FILE DOWNLOADED AND PEER DOWNLOAD COUNT UPDATED TO 1 .

```

Tracker:

```

[{'username': 1, 'password': 1, 'count_downloads': 0, 'count_failures': 0},
[{'username': 2, 'password': 2, 'count_downloads': 0, 'count_failures': 0},
password: 1, 'count_downloads': 0, 'count_failures': 0, 'count_successful_downloads': 0},
New connection from ('127.0.0.1', 49210)
File: file6.txt Available from {229, 135}

```

UNSUCCESSFULL DOWNLAOD(file does not exist)

```

Enter your choice (1-5): 2

Enter the filename to download: file6.txt
Error retrieving the details of the file: FILE DOES NOT EXIST

```

UNSUCCESSFULL DOWNLOAD(το παράγουμε συνθετικά στο αποδέκτη peer δημιουργώντας timeout exception)

Peer.py

```

208         for p in peers:
209             current_peer= p.token_id
210             try:
211                 # Uncomment to simulate download failure
212                 raise socket.timeout("Server did not respond within the timeout period.")
213

```

5. Exit

Enter your choice (1-5): 2

Enter the filename to download: file1.txt

FILE DOWNLOAD FAILED AND PEER DOWNLOAD FAILURE COUNT UPDATED TO 1

Peer Menu:

General

Τα sockets στην python γενικά είναι πολύ απλά αλλά τα errors , μερικές φορές, δύσκολα να βρούμε τη πηγή τους γιατί είναι dynamically typed. Το δύσκολο ήταν η διαχείριση όλων των δομών για να είναι συνεπές με το τί συμβαίνει στο δίκτυο των