Πανεπιστήμιο Αιγαίου
Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Κατανεμημένα Συστήματα - 2η Ομαδική Εργασία
Ημερομηνία Παράδοσης : 25/05/2023

Εργαστηριακοί Διδάσκοντες : Δούμα Αναστασία

ΔΕΥΤΕΡΗ ΟΜΑΔΙΚΗ ΕΡΓΑΣΙΑ

Θέμα – Ανάπτυξη κατανεμημένου συστήματος με Java Sockets και RMI

ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ

Το κατανεμημένο σύστημα που θέλουμε να αναπτύξουμε ακολουθεί το μοντέλο πελάτη-εξυπηρετητή-εξυπηρετητή και αφορά μια απλοποιημένη θεώρηση ενός συστήματος για κράτηση αεροπορικών εισιτηρίων. Το σύστημα αποτελείται από δύο εξυπηρετητές και πολλούς πελάτες που αντιπροσωπεύουν τους πιθανούς αγοραστές αεροπορικών εισιτηρίων.

Πιο συγκεκριμένα, οι πελάτες θα υποβάλουν το αίτημα τους μέσω μιας γραφικής διεπαφής (GUI) στον εξυπηρετητή 1. Ο εξυπηρετητής 1 είναι υπεύθυνος για να δέχεται όλα τα αιτήματα από τους υποψήφιους πελάτες, και αφού πρώτα επικοινωνήσει με τον εξυπηρετητή 2, ο οποίος αποτελεί μια βάση με όλες τις πληροφορίες για τις διαθέσιμες πτήσεις, να τους ενημερώνει για την διαθεσιμότητα των πτήσεων. Σε περίπτωση που γίνει προσπάθεια κράτησης εισιτηρίων από κάποιο πελάτη, ο εξυπηρετητής 1 είναι υπεύθυνος να ενημερώσει τον πελάτη για το εάν μπόρεσε να γίνει η κράτηση ή όχι εφόσον ανταλλάξει σχετικά μηνύματα με τον εξυπηρετητή 2. Οι τεχνολογίες που θα χρησιμοποιηθούν για την δημιουργία του συστήματος είναι Java RMI για την σύνδεση των πελατών με τον εξυπηρετητή 1 και sockets ή datagrams για την σύνδεση του εξυπηρετητή 1 με τον εξυπηρετητή 2.

Πελάτες

Οι πελάτες θα πρέπει να μπορούν να κάνουν αναζητήσεις για διαθέσιμες πτήσεις. Για αναζήτηση θα μπορούν να θέτουν τις εξής πληροφορίες:

Από: Σάμο
Προς: Αθήνα
Αριθμός Επιβατών: 2
Ημέρα αναχώρησης: 2 Μαΐου
Ημέρα επιστροφής: 9 Μαΐου

Οι πελάτες αφού θα καταχωρήσουν τις πληροφορίες για το ταξίδι που θέλουν να αναζητήσουν, θα ενημερώνονται από τον εξυπηρετητή 1 εάν υπάρχουν διαθέσιμες πτήσεις ή όχι. Σε περίπτωση που υπάρχουν, θα εμφανίζονται στον χρήστη και θα του δίνεται η επιλογή να επιλέξει την πτήση που επιθυμεί να κάνει κράτηση ή να ξανά κάνει αναζήτηση. Για την ίδια μέρα αναχώρησης ή επιστροφής, θα μπορούν να υπάρχουν παραπάνω από μια πτήσεις για τον ίδιο προορισμό, αλλά σε διαφορετικές ώρες. Σε περίπτωση που ο χρήστης επιλέξει να κάνει κράτηση, θα πρέπει να περιμένει από τον εξυπηρετητή 1 να τον ενημερώσει εάν τελικά έγινε ή όχι η κράτηση. Σε περίπτωση που δεν υπάρχουν διαθέσιμες πτήσεις ή δεν γίνει η κράτηση θα δίνεται η επιλογή στον χρήστη να ξανά κάνει αναζήτηση.

1ος Εξυπηρετητής
Ο εξυπηρετητής 1 θα μπορεί να εξυπηρετεί παραπάνω από έναν πελάτες ταυτόχρονα. Είναι υπεύθυνος για να δέχεται τα ερωτήματα για όλα τα διαθέσιμα δρομολόγια από τους πελάτες, και αφού πρώτα επικοινωνήσει με τον εξυπηρετητή 2, να τους ενημερώνει για τις διαθέσιμες πτήσεις. Επιπλέον είναι υπεύθυνος για να δεχτεί αίτημα κράτησης από τους πελάτες, να αποφασίσει εάν μπορεί να γίνει η κράτηση ή ποια κράτηση θα πραγματοποιηθεί σε συνεργασία με τον εξυπηρετητή 2, και να ενημερώσει τον πελάτη για την εξέλιξη της κράτησης τους.

2ος Εξυπηρετητής
Ο εξυπηρετητής 2 αποτελεί μια βάση με όλες τις πληροφορίες για τις διαθέσιμες πτήσεις. Κάποιες από τις πληροφορίες που θα διαθέτει για κάθε πτήση είναι:

Ημερομηνία, Ώρα, Από, Προς, Αριθμός Πτήσης, Αριθμός θέσεων, Τιμή Εισιτηρίου

Ο εξυπηρετητής 2 αφού λάβει ένα ερώτημα από τον εξυπηρετητή 1 θα κάνει αναζήτηση στη βάση και θα του απαντήσει με τα διαθέσιμα αποτελέσματα. Επίσης είναι υπεύθυνος να ενημερώνει τη βάση του σε περίπτωση που υπάρχει κράτηση εισιτηρίου. Η βάση με τις πληροφορίες θα μπορούσε να είναι: SQLite database ή άλλη αντίστοιχη ΒΔ.

Συνθήκες ανταγωνισμού (Race Conditions)
Ένα κατανεμημένο σύστημα σαν αυτό της άσκησης ευνοεί την εμφάνιση συνθηκών ανταγωνισμού. Περιγράψτε και υλοποιήστε ένα σενάριο για το σύστημα που να δείχνει την εμφάνιση συνθηκών ανταγωνισμού. Πώς μπορούμε να αντιμετωπίσουμε αυτό το πρόβλημα; Χρησιμοποιήστε μηχανισμούς που διαθέτει η Java, για να αντιμετωπίσετε το πρόβλημα.

Αρχιτεκτονική/Τεχνολογία Υλοποίησης
Ποια θεωρείται ότι είναι τα πλεονεκτήματα της αρχιτεκτονικής (πελάτης - εξυπηρετητής - εξυπηρετητής) με την οποία υλοποιείται η συγκεκριμένη κατανεμημένη εφαρμογή έναντι του απλού μοντέλου πελάτη/εξυπηρετητή; Στην παρούσα εργασία χρησιμοποιείται RMI και Sockets για την διασύνδεση των οντοτήτων του συστήματος σας. Ποια θεωρείτε ότι είναι τα πλεονεκτήματα της υλοποίησης της εφαρμογής με RMI έναντι της υλοποίησης με Sockets;

Ασφάλεια
Ένα ζήτημα σε μια ρεαλιστική υλοποίηση της συγκεκριμένης εφαρμογής είναι η διαχείριση της ασφάλειας του συστήματος κατά την αλληλεπίδραση πελάτη - εξυπηρετητή αλλά και στην επικοινωνία μεταξύ των δύο εξυπηρετητών. Περιγράψτε μηχανισμούς που μπορούν να χρησιμοποιηθούν για την προστασία του συστήματος. Το ζητούμενο αυτό δεν θα πρέπει να υλοποιηθεί στην τρέχουσα έκδοση της εφαρμογής.

Bonus: Ορίστε πολιτικές ασφαλείας και για τις 3 οντότητες της εφαρμογής που υλοποιείτε. Δηλαδή για τον 1ο εξυπηρετητή, τον 2ο εξυπηρετητή και για τον πελάτη.

Γενικές Υποδείξεις
- Η εργασία μπορεί να εκπονηθεί από ομάδα φοιτητών (1 ή 2 φοιτητές).

- Η υλοποίηση θα περιλαμβάνει τρία διαφορετικά projects, το πρόγραμμα του πελάτη και τα προγράμματα των εξυπηρετητών.
- Οι πελάτες επικοινωνούν κάνοντας χρήση Java RMI, ενώ οι εξυπηρετητές χρησιμοποιώντας Java sockets για τη διαδιεργασιακή επικοινωνία.
- Θα χρειαστεί να ορίσετε ένα request/reply πρωτόκολλο επικοινωνίας μεταξύ του πελάτη, του εξυπηρετητή 1 και του εξυπηρετητή 2 και τη μορφή των μηνυμάτων που ανταλλάσσονται.
- Θα χρειαστεί να αποφασίσετε το είδος της σύνδεσης μεταξύ εξυπηρετητή 1 και εξυπηρετητή 2, δηλαδή χρήση πρωτοκόλλου μετάδοσης δικτύου προσανατολισμένο σε σύνδεση (connection-oriented), ή πρωτοκόλλου άνευ σύνδεσης (connectionless).
- Ο χρήστης επιλέγει την λειτουργία που επιθυμεί μέσα από ένα απλό μενού που παρουσιάζεται στην οθόνη. Μόνο για τους πελάτες θα πρέπει υλοποιήσετε γραφική διεπαφή.
- Εφόσον δεν έχετε δίκτυο υπολογιστών ώστε να τρέξετε τις διεργασίες πελάτη και εξυπηρετητή σε διαφορετικούς κόμβους μπορείτε να χρησιμοποιήσετε την διεύθυνση localhost (IP address 127.0.0.1) η οποία δίνει την ψευδαίσθηση ενός δικτύου με έναν κόμβο. Ανοίξτε διαφορετικά παράθυρα τερματικών για τον πελάτη και τους εξυπηρετητές.
- Διασφαλίστε ότι κάθε φορά που μια σύνδεση με τον εξυπηρετητή τερματίζεται απελευθερώνονται οι πόροι του συστήματος (δυναμική μνήμη, socket descriptors, κ.α.).

Οδηγίες για παράδοση Ασκήσεων:

- Η παράδοση των εργασιών θα γίνει **ηλεκτρονικά** μέσω του **e-class** μέχρι την **Πέμπτη 25/ 05/2023** στις **23.59** το βράδυ. Εκπρόθεσμη υποβολή μέχρι μια εβδομάδα χάνει το 25% της βαθμολογίας, ενώ πάνω από μια εβδομάδα δεν γίνεται δεκτή.
- Μόνο το ένα μέλος της ομάδας θα πρέπει να υποβάλλει την εργασία με μορφή συμπιεσμένου αρχείου zip ή rar. Το όνομα του αρχείου θα είναι: **DSPROJECT02icsdxxxxx.<rar|zip>**. Στο αρχείο θα περιέχονται τα εξής:
  - Τα 3 project (εξυπηρετητές και πελάτης) από το Netbeans με τα αρχεία της εφαρμογής (Στην αρχή του πηγαίου κώδικα κάθε κλάσης θα αναγράφεται ο αριθμός μητρώου και το ονοματεπώνυμο σας). ***Επιβάλλεται η χρήση σχολίων στον κώδικα σας***.

  - Ένα αρχείο .pdf με τα ακόλουθα :

  1. Τις βασικές σχεδιαστικές αποφάσεις που έχετε λάβει για την υλοποίηση της εφαρμογής.
  2. Τυχόν παραδοχές που έχετε κάνει, πέραν των προδιαγραφών που σας έχουν δοθεί.
  3. Το request/reply πρωτόκολλο επικοινωνίας που ορίσατε μεταξύ του πελάτη, του εξυπηρετητή 1 και του εξυπηρετητή 2 και τη μορφή των μηνυμάτων που ανταλλάσσονται (με ένα σχετικό διάγραμμα).
  4. Οδηγίες για την εκτέλεση τόσο της εφαρμογής του πελάτη όσο και της εφαρμογής κάθε εξυπηρετητή και
  5. Οθόνες εκτέλεσης της εφαρμογής με διαφορετικά σενάρια.

Στην αρχή του αρχείου θα αναγράφεται ο αριθμός μητρώου και το ονοματεπώνυμο σας.

- **Καμία εργασία δεν θα διορθωθεί εάν δεν έχει ακριβώς αυτή τη μορφή.**

- Η υποβολή κοινών απαντήσεων από διαφορετικούς φοιτητές δεν επιτρέπεται και θεωρείται ως **ΑΝΤΙΓΡΑΦΗ**. Η αντιγραφή έχει ως αποτέλεσμα το ΜΗΔΕΝΙΣΜΟ ΤΗΣ ΕΡΓΑΣΙΑΣ ΣΥΝΟΛΙΚΑ.