



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Εργασία Auction Server

System Documentation

Βάσιλας Δημήτριος	AM:03110660
Βραχασωτάκης Μιχάλης	AM:03110061

Επιλέξαμε να αναπτύξουμε την εφαρμογή με χρήση της γλώσσας προγραμματισμού Python, λόγω του ότι είναι μία ευρέως χρησιμοποιούμενη, ευέλικτη γλώσσα, η οποία διαθέτει όλη τη λειτουργικότητα που χρειαζόμαστε για την υλοποίηση της εφαρμογής (threading, sockets, σύνδεση με DBMS).

Οι αρχιτεκτονικές τεχνολογίες με τις οποίες σχεδιάστηκε και υλοποιήθηκε το σύστημα είναι Server-Client κατά την αλληλεπίδραση μεταξύ των auctioneers και bidders, και Peer-To-Peer κατά την αλληλεπίδραση μεταξύ των auctioneers.

Auctioneers

Ο auctioneer server είναι η κλάση η οποία χειρίζεται την σύνδεση των χρηστών στο σύστημα και διαχειρίζεται την διαδικασία της δημοπρασίας. Στο παρών σύστημα υπάρχουν δύο auctioneer servers, οι οποίοι στην υλοποίηση μας εκτελούνται ως δύο thread τις ίδιες διεργασίας. Το κύριο πλεονέκτημα της επιλογής αυτής είναι ότι αποφεύγεται το κόστος και η πολυπλοκότητα της διαδιεργασιακής επικοινωνίας.

Για τη σύνδεση των bidders στο σύστημα και την επικοινωνία με τους auctioneers χρησιμοποιείται η δομή των sockets. Οι auctioneers κατά την εκκίνηση τους δημιουργούν από ένα socket το οποίο περιμένει εισερχόμενες συνδέσεις σε κάποια προκαθορισμένη θύρα, και ανακοινώνουν αυτή την θύρα εγγράφοντας την σε ένα αρχείο ώστε οι bidders να μπορούν να χρησιμοποιήσουν την πληροφορία αυτή για να συνδεθούν σε έναν από τους δύο auctioneers.

Η χρήση sockets είναι χρήσιμη και για την διαχείριση των συνδεδεμένων χρηστών από κάθε auctioneer. Οι auctioneer servers κρατούν σε δομές τους file descriptors των συνδεδεμένων bidders και διαχειρίζονται την επικοινωνία χρησιμοποιώντας την κλήση select() η οποία μπλοκάρει την εκτέλεση του προγράμματος έως ότου κάποιος από του file descriptors να έχει διαθέσιμα δεδομένα.

Η λειτουργία του συστήματος βασίζεται στην ανταλλαγή μηνυμάτων μεταξύ των δύο πλευρών που συμμετέχουν στη δημοπρασία. Στη συνέχεια αναφέρουμε τους τύπους μηνυμάτων που στέλνονται από τους auctioneers στους bidders. Στην επόμενη ενότητα θα αναφέρουμε και τα αντίστοιχα μηνύματα με αντίθετη κατεύθυνση.

- **bid_item**: ενημερώνει τους bidders για το επόμενο αντικείμενο προς δημοπράτηση. Τα στοιχεία που αποστέλλονται είναι η περιγραφή και η αρχική τιμή του αντικειμένου.
- **start_bidding**: ενημερώνει τους bidders που έχουν δηλώσει ενδιαφέρον ότι αρχίζει η διαδικασία της δημοπρασίας για το συγκεκριμένο αντικείμενο, και μπορούν να υποβάλλουν προσφορές.
- **new_high_bid**: ενημερώνει τους bidders που συμμετέχουν στη δημοπράτηση ενός αντικειμένου για μία νέα μέγιστη προσφορά για το αντικείμενο. Αποστέλλεται η νέα μέγιστη προσφορά όχι όμως και το username του χρήστη που την έκανε για λόγους ασφάλειας.
- **price_drop**: ενημερώνει τους bidders για την μείωση της αρχικής τιμής του αντικειμένου λόγω απουσίας προσφορών. Αποστέλλεται η νέα αρχική τιμή του αντικειμένου.
- **stop_bidding**: ενημερώνει τους bidders ότι η δημοπράτηση του αντικειμένου έχει ολοκληρωθεί και δε γίνονται πλέον δεκτές νέες προσφορές. Το μήνυμα περιλαμβάνει το username του bidder στον οποίο πιστώνεται το αντικείμενο.
- **budget**: Το μήνυμα στέλνεται στον bidder που έκανε την μεγαλύτερη προσφορά σε μία δημοπρασία, μετά το τέλος αυτής, και τον ενημερώνει για το νέο του διαθέσιμο ποσό μετά την αφαίρεση της πληρωμής για το αντικείμενο.

- `action_complete`: Ενημερώνει όλους του συνδεδεμένους χρήστες για την ολοκλήρωση της διαδικασίας ώστε να τερματίσουν τη λειτουργία τους.
- `duplicate_name`: Ενημερώνει τον bidder που προσπαθεί να συνδεθεί για την ύπαρξη άλλου χρήστη με το ίδιο username σε κάποιον από τους δύο auctioneers, ώστε αυτός να τερματιστεί με κατάλληλο μήνυμα.

Οι auctioneers χρησιμοποιούν ο καθένας μία διαφορετική βάση δεδομένων για να αποθηκεύουν τα στοιχεία των συνδεδεμένων χρηστών, των αντικειμένων προς δημοπράτηση, καθώς και των ολοκληρωμένων δημοπρατήσεων. Οι λεπτομέρειες για τη χρήση των βάσεων δεδομένων περιγράφονται παρακάτω.

Η ύπαρξη δύο διαφορετικών auctioneers που χρησιμοποιούν ξεχωριστές βάσεις δεδομένων δημιουργεί την ανάγκη συγχρονισμού μεταξύ τους, για ζητήματα όπως την ενημέρωση για σύνδεση νέων χρηστών και την ύπαρξη νέας μέγιστης προσφοράς για κάποιο αντικείμενο, αλλά και ζητήματα ασφαλείας όπως τον έλεγχο ύπαρξης διπλών ονομάτων μεταξύ των δύο auctioneers. Ο συγχρονισμός επιτυγχάνεται με τη χρήση ενός απλού gossip αλγορίθμου, με τη βοήθεια ενός synchronisation server, όπου κάθε auctioneer ανακοινώνει στον άλλο όποτε υπάρχει ένα νέο γεγονός. Σημειώνεται ότι για να αντιμετωπιστεί η περίπτωση όπου κάποιος από τους δύο auctioneers καθυστερήσει για κάποιο λόγο και ο άλλος τελειώσει κάποιο countdown νωρίτερα τότε ο δεύτερο περιμένει τον πρώτο να φτάσει στο ίδιο σημείο ώστε να συνεχίσουν συγχρόνως, παραμένοντας όμως ανενεργός ώστε να μη δίνει στους χρήστες του περισσότερο από τον προβλεπόμενο χρόνο. Το πρόβλημα όπου ο ένας auctioneer σταματήσει να δέχεται προσφορές πριν από τον άλλο, και υπάρξει μία νέα μέγιστη προσφορά αντιμετωπίζεται συγκρίνοντας τις μέγιστες προσφορές όταν φτάσουν στο ίδιο σημείο και ανακοινώνοντας κοινό νικητή. Η υλοποίηση του συγχρονισμού θα αναλυθεί περισσότερο παρακάτω.

Για την υλοποίηση του auctioneer server χρησιμοποιήθηκαν τα παρακάτω design patterns:

- **Multiton**: μία παραλλαγή του singleton που επιτρέπει τη δημιουργία παραπάνω του ενός instances μίας κλάσης. Εδώ χρειαζόμαστε την ύπαρξη δύο auctioneers οπότε χρησιμοποιούμε multiton για δύο instances.
- **Iterator**: χρησιμοποιείται συχνά στην υλοποίηση μας για να έχουμε πρόσβαση σε όλα τα στοιχεία μίας δομής διαδοχικά χωρίς να γνωρίζουμε λεπτομέρειες για την υλοποίηση της.

Bidders

Ο Bidder είναι η κλάση η οποία είναι υπεύθυνη για την υπόλοιπη λειτουργικότητα του συστήματος. Αναλαμβάνει την διαδραστική επικοινωνία με τον χρήστη και τη μεταφορά των αποφάσεων του στον auctioneer.

Χρησιμοποιεί, αντίστοιχα με τον auctioneer, sockets για την επικοινωνία, και την εντολή select για να τίθεται σε λειτουργία όταν υπάρχουν δεδομένα είτε στο socket είτε στο standard input από τον χρήστη, και να εκτελεί την αντίστοιχη λειτουργία.

Τα μηνύματα που στέλνονται μέσω του socket προς τον auctioneer στον οποίο είναι συνδεδεμένος ο bidder είναι:

- connect: ο bidder ανακοινώνει την σύνδεση του στον auctioneer. Αποστέλλεται μαζί με το username και το συνολικό διαθέσιμο ποσό για αγορές.
- i_am_interested: δήλωση ενδιαφέροντος για συμμετοχή στη δημοπρασία για το παρών αντικείμενο.
- bid: η ανακοίνωση για μία νέα προσφορά από τον χρήστη για το τρέχον αντικείμενο, μαζί με το ποσό της προσφοράς.
- quit: ο bidder πρόκειται να τερματιστεί και ειδοποιεί τον auctioneer να πράξει όλες τις απαιτούμενες ενέργειες για τερματισμό της σύνδεσης.

Για την υλοποίηση του bidder χρησιμοποιήθηκαν τα παρακάτω design patterns:

- Iterator: για την απλοποίηση της διαδοχικής πρόσβασης στα αντικείμενα μίας δομής.

Synchronisation Server

Ο Synchronisation server είναι η κλάση η οποία αναλαμβάνει τον συγχρονισμό της λειτουργίας των δύο auctioneers. Η λειτουργία του είναι αρκετά απλή. Για κάθε λειτουργία που απαιτεί συντονισμό, όταν ο ένας auctioneer χρειάζεται να ενημερώσει τον άλλο για κάποιο συμβάν, καλεί την αντίστοιχη μέθοδο του synchronisation server, ο οποίος ειδοποιεί τον άλλο auctioneer καλώντας μία δικιά του κατάλληλη μέθοδο. Με αυτόν τον τρόπο επιτυγχάνεται ο η επικοινωνία μεταξύ των auctioneer servers. Ο συγχρονισμός γίνεται με δύο λειτουργίες: Πρώτον ο ένας auctioneer ενημερώνει τον άλλο για σημαντικά συμβάντα, και δεύτερον σε κάποιες περιπτώσεις ο ένας auctioneer περιμένει ειδοποίηση από τον άλλο για να συνεχίσει τη λειτουργία του.

Για την υλοποίηση του synchronisation server χρησιμοποιήθηκαν τα παρακάτω design patterns:

- Observer: χρησιμοποιήθηκε το σχεδιαστικό μοτίβο observer ώστε ο ένας auctioneer να γνωστοποιεί στον άλλο αλλαγές στην κατάσταση του, μέσω του synchronisation server.
- Adapter: χρησιμοποιήθηκε ώστε ο κώδικας των auctioneers να καλεί τη μέθοδο του synchronisation server που επιτυγχάνει τον συγχρονισμό, χωρίς να γνωρίζει λεπτομέρειες για τη διαπροσωπεία της

Driver Script

Το script αυτό αναλαμβάνει να πραγματοποιήσει όλη τη διαδικασία αυτόματα και να καταγράψει το log σε ένα αρχείο. Εκτελείτε με ορίσματα configuration files (περιγράφονται στο έγγραφο user documentation), τα οποία ορίζουν το πόσοι bidders θα δημιουργηθούν, την συχνότητα και τα ποσά με την οποία θα πραγματοποιούν προσφορές. Εκκινεί τους auctioneers και τον κατάλληλο αριθμό bidders και πραγματοποιεί τις ενέργειες που περιγράφονται στα configuration files, εκ μέρους των χρηστών.

Βάσεις Δεδομένων

Για τη χρήση βάσεων δεδομένων στην εφαρμογή επιλέξαμε τη χρήση του συστήματος διαχείρισης βάσεων δεδομένων MySQL, λόγω του εύχρηστου API που διαθέτει και την υποστήριξη σε μεγάλο εύρος λειτουργικών συστημάτων.

Οι πίνακες που χρησιμοποιούνται στην εφαρμογή είναι οι εξής:

- Users: στον πίνακα αυτό αποθηκεύονται τα στοιχεία των συνδεδεμένων στο σύστημα χρηστών. Τα πεδία του πίνακα είναι τα εξής:
 - Id : ένα μοναδικό αναγνωριστικό για κάθε χρήστη (η αύξουσα σειρά σύνδεσης του στο σύστημα).
 - Username: το username το χρήστη.
 - Budget: το διαθέσιμο ποσό του χρήστη για αγορές.
- Items: στον πίνακα αυτό αποθηκεύονται τα στοιχεία των αντικειμένων που πρόκειται να δημοπρατηθούν. Τα πεδία του πίνακα είναι τα εξής
 - Id: ένα μοναδικό αναγνωριστικό για κάθε αντικείμενο.
 - Description: η περιγραφή του αντικειμένου.
 - Price: η αρχική τιμή.
- Bought: στον πίνακα αυτό αποθηκεύονται τα στοιχεία των ολοκληρωμένων δημοπρατήσεων αντικειμένων. Τα πεδία του πίνακα είναι τα εξής:
 - UserId: το Id του χρήστη που αγόρασε το αντικείμενο.
 - ItemId: το Id του αντικειμένου που αγοράστηκε.
 - Amount: το ποσό για το οποίο ο χρήστης αγόρασε το αντικείμενο.

Παρακάτω παρουσιάζονται τα SQL script για τη δημιουργία των πινάκων αυτών.

```
CREATE TABLE `Bought` (  
  `UserId` int(11) NOT NULL,  
  `ItemId` int(11) NOT NULL,  
  `Amount` float NOT NULL,  
  PRIMARY KEY (`UserId`,`ItemId`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `Items` (  
  `Id` int(11) NOT NULL,  
  `Description` varchar(200) NOT NULL,  
  `Price` float NOT NULL,  
  PRIMARY KEY (`Id`),  
  UNIQUE KEY `Id_UNIQUE` (`Id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `Users` (  
  `Id` int(11) NOT NULL,  
  `Username` varchar(45) NOT NULL,  
  `Budget` float NOT NULL,  
  PRIMARY KEY (`Id`),  
  UNIQUE KEY `Id_UNIQUE` (`Id`),  
  UNIQUE KEY `Username_UNIQUE` (`Username`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Παρακάτω παρουσιάζεται η καταγραφή μίας εκτέλεσης του συστήματος μέσω του driver server, η οποία καλύπτει αρκετές περιπτώσεις χρήσης:

```
(MainThread) Socket Connected to ip 127.0.0.1  
(auctioneer1) Connected with 127.0.0.1 :59729  
(MainThread) Socket Connected to ip 127.0.0.1  
(auctioneer2) Connected with 127.0.0.1 :47366  
(a ) The name you entered already exists. Shutting down  
(MainThread) Socket Connected to ip 127.0.0.1  
(auctioneer1) Connected with 127.0.0.1 :59736  
(MainThread) Socket Connected to ip 127.0.0.1  
(auctioneer2) Connected with 127.0.0.1 :47372  
(MainThread) Socket Connected to ip 127.0.0.1  
(auctioneer2) Connected with 127.0.0.1 :47373  
(a ) The next item is:  
      A brand-new zodiac watch  
and it has a starting price of:  
      200.0  
(a ) Press <Enter> to enter auction or wait for next item  
(b ) The next item is:  
      A brand-new zodiac watch  
and it has a starting price of:  
      200.0  
(b ) Press <Enter> to enter auction or wait for next item  
(d ) The next item is:  
      A brand-new zodiac watch  
and it has a starting price of:  
      200.0  
(c ) The next item is:  
      A brand-new zodiac watch  
and it has a starting price of:  
      200.0  
(d ) Press <Enter> to enter auction or wait for next item  
(c ) Press <Enter> to enter auction or wait for next item  
(auctioneer1) a joined the auction  
(auctioneer1) b joined the auction  
(auctioneer2) c joined the auction
```

(auctioneer2) bidding has started
(auctioneer2) initial price 200.0
(c) bidding has started
(c) initial price 200.0
(c) Options
1: item discription
2: highest bid
3 <amount>: new bid
Available budget 500000.0
(auctioneer1) bidding has started
(auctioneer1) initial price 200.0
(a) bidding has started
(a) initial price 200.0
(a) Options
1: item discription
2: highest bid
3 <amount>: new bid
Available budget 1000000.0
(b) bidding has started
(b) initial price 200.0
(b) Options
1: item discription
2: highest bid
3 <amount>: new bid
Available budget 999999.0
(auctioneer1) new bid: amount= 200.0
(a) new high bid
(a) amount 200.0
(c) new high bid
(b) new high bid
(c) amount 200.0
(b) amount 200.0
(auctioneer1) new bid: amount= 200.1
(a) new high bid
(a) amount 200.1
(c) new high bid
(b) new high bid
(c) amount 200.1
(b) amount 200.1
(auctioneer2) new bid: amount= 201.0
(a) new high bid
(c) new high bid
(a) amount 201.0
(b) new high bid
(c) amount 201.0
(b) amount 201.0
(auctioneer2) bidding has stopped
(auctioneer1) bidding has stopped

- (c) bidding has stopped
- (c) highest bidder c
- (c) item bought
- (c) waiting for next item
- (a) bidding has stopped
- (b) bidding has stopped
- (a) highest bidder c
- (b) highest bidder c
- (a) waiting for next item
- (b) waiting for next item
- (c) The next item is:

The one and only iPhone6.0000001 in the world
and it has a starting price of:

10000.0

- (d) The next item is:

The one and only iPhone6.0000001 in the world
and it has a starting price of:

10000.0

- (c) Press <Enter> to enter auction or wait for next item
- (d) Press <Enter> to enter auction or wait for next item
- (a) The next item is:

The one and only iPhone6.0000001 in the world
and it has a starting price of:

10000.0

- (a) Press <Enter> to enter auction or wait for next item
- (b) The next item is:

The one and only iPhone6.0000001 in the world
and it has a starting price of:

10000.0

- (b) Press <Enter> to enter auction or wait for next item
- (auctioneer1) a joined the auction
- (auctioneer1) bidding has started
- (auctioneer1) initial price 10000.0

- (a) bidding has started
- (a) initial price 10000.0
- (a) Options

1: item discription

2: highest bid

3 <amount>: new bid

Available budget 1000000.0

- (a) price dropped
- (a) new amount 9000.0
- (a) price dropped
- (a) new amount 8100.0
- (a) price dropped
- (a) new amount 7290.0
- (a) price dropped
- (a) new amount 6561.0

(a) price dropped
(a) new amount 5904.9
(auctioneer1) new bid: amount= 6000.0
(a) new high bid
(a) amount 6000.0
(auctioneer1) bidding has stopped
(a) bidding has stopped
(a) highest bidder a
(a) item bought
(a) waiting for next item
(auctioneer2) bidding has stopped
(b) The next item is:
 A Lenovo-Gk015
and it has a starting price of:
 3000.0
(a) The next item is:
 A Lenovo-Gk015
and it has a starting price of:
 3000.0
(b) Press <Enter> to enter auction or wait for next item
(a) Press <Enter> to enter auction or wait for next item
(c) The next item is:
 A Lenovo-Gk015
and it has a starting price of:
 3000.0
(d) The next item is:
 A Lenovo-Gk015
and it has a starting price of:
 3000.0
(c) Press <Enter> to enter auction or wait for next item
(d) Press <Enter> to enter auction or wait for next item
(auctioneer1) a joined the auction
(auctioneer1) b joined the auction
(auctioneer1) bidding has started
(auctioneer1) initial price 3000.0
(a) bidding has started
(b) bidding has started
(a) initial price 3000.0
(b) initial price 3000.0
(a) Options
1: item discription
2: highest bid
3 <amount>: new bid
Available budget 994000.0
(b) Options
1: item discription
2: highest bid
3 <amount>: new bid

Available budget 999999.0
(auctioneer1) new bid: amount= 3000.0
(a) new high bid
(b) new high bid
(a) amount 3000.0
(b) amount 3000.0
(auctioneer1) new bid: amount= 3100.0
(b) new high bid
(b) amount 3100.0
(a) new high bid
(a) amount 3100.0
(auctioneer1) new bid: amount= 3110.0
(a) new high bid
(a) amount 3110.0
(b) new high bid
(b) amount 3110.0
(auctioneer1) bidding has stopped
(a) bidding has stopped
(b) bidding has stopped
(a) highest bidder a
(b) highest bidder a
(a) item bought
(b) waiting for next item
(a) waiting for next item
(auctioneer2) bidding has stopped
(b) The next item is:
 A left-handed ukulele
and it has a starting price of:
 80.0
(b) Press <Enter> to enter auction or wait for next item
(a) The next item is:
 A left-handed ukulele
and it has a starting price of:
 80.0
(a) Press <Enter> to enter auction or wait for next item
(c) The next item is:
 A left-handed ukulele
and it has a starting price of:
 80.0
(d) The next item is:
 A left-handed ukulele
and it has a starting price of:
 80.0
(c) Press <Enter> to enter auction or wait for next item
(d) Press <Enter> to enter auction or wait for next item
(auctioneer2) nobody interested, discarding item
(auctioneer1) bidding has stopped
(auctioneer2) no more items. Exiting...

(c) The auction has been completed. Thank you for participating

(d) The auction has been completed. Thank you for participating

(auctioneer1) no more items. Exiting...

(a) The auction has been completed. Thank you for participating

(b) The auction has been completed. Thank you for participating