

MailServer

Εργασία Δικτυακού Προγραμματισμού

Ημερομηνία: 07/01/21

Υπεύθυνος Καθηγητής: Παπαδημητρίου Γιώργος

Υπεύθυνος Φοιτητής: Βασιλειάδης Γιώργος

AEM: 3466

Στην παρούσα αναφορά καλύπτονται τα εξής θέματα:

- Γενικός Σχολιασμός της Εργασίας
- Διασαφήνιση των Βασικών Στοιχείων
- Επεξήγηση Κώδικα
- Επεξήγηση Χρήσης
- Εποπτικά Παραδείγματα Χρήσης

Γενικός Σχολιασμός της Εργασίας

Η παρούσα εργασία αποτελεί προϊόν αποκλειστικής προσπάθειας του υπεύθυνου φοιτητή. Δεν χρησιμοποιήθηκαν έτοιμες βιβλιοθήκες τρίτων, πέραν της βασικής που παρέχεται με τη γλώσσα Java.

Για την παραγωγή των αρχείων .jar χρησιμοποιήθηκε η όγδοη έκδοση της γλώσσας Java (Java 8) και αυτή αποτελεί την ελάχιστη απαίτηση για την ορθή λειτουργία του προγράμματος.

Ο κώδικας είναι επαρκώς σχολιασμένος. Ωστόσο, παρέχονται επιπλέον επεξηγήσεις στις μετέπειτα ενότητες. Σε αυτές δίνεται μια ευρύτερη εικόνα της λειτουργίας του κώδικα ως ένα σύνολο υπομονάδων.

Κρίνεται ότι η παρούσα εργασία ανταποκρίνεται σε όλες τις απαιτήσεις που δόθηκαν μέσω του υπεύθυνου καθηγητή και του σχετικού .pdf οδηγιών που επισυνάπτεται.

Οι εκτελέσιμες μορφές του παραδοτέου βρίσκονται υπό τον φάκελο “jar”. Ο πηγαίος κώδικας μπορεί να βρεθεί στον φάκελο “src”.

Για λόγους παρουσίας δυνατοτήτων, υπάρχουν τρεις προ-εγγεγραμμένοι χρήστες, με τρία εισερχόμενα, αδιάβαστα μηνύματα στον λογαριασμό του καθενός. Τα στοιχεία τους βρίσκονται στο ανώτερο επίπεδο του παραδοτέου, στο αρχείο “accounts.txt”.

Οι έλεγχοι εισόδου του χρήστη περιορίστηκαν στους απολύτως απαραίτητους. Για αυτόν τον λόγο υπάρχουν ορισμένες μη-κριτικές ασυνέπειες (λόγου χάρη, ένας χρήστης μπορεί να έχει κενό όνομα, ή κενό κωδικό). Σε κάθε περίπτωση, το σύστημα παραμένει σε κατάσταση εγκυρότητας (στο παράδειγμα, ο χρήστης με το κενό όνομα, είναι μοναδικός κ.ο.κ).

Κατά τη σύνταξη ενός e-mail ο χρήστης ενδέχεται να χρειάζεται παραπάνω από μία γραμμή στο κυρίως μήνυμά του. Έτσι, όταν συμπληρώνει το πεδίο του κυρίως μηνύματος, το σύστημα δέχεται κάθε νέα είσοδό που παρέχει ο χρήστης ως μέρος του μηνύματος. Για να βγει από αυτή την κατάσταση, ο χρήστης χρειάζεται να πληκτρολογήσει και να εισάγει σε κενή γραμμή την ακολουθία “<ok>”.

Διασαφήνιση των Βασικών Στοιχείων

Είναι εμφανές ότι το μοντέλο που χρησιμοποιήθηκε για την εκπόνηση της εργασίας είναι αυτό του πελάτη-εξυπηρετητή (στο εξής client-server). Καθένα από τα δύο μέρη έχει την ευθύνη για τη δημιουργία μηνυμάτων, την αποστολή τους, και την επεξεργασία των εισερχόμενων μηνυμάτων. Όλα τα μηνύματα μεταξύ server και client γίνονται μέσω καθαρού κειμένου (plain-text). Συνεπώς, το πακετάρισμα και το ξεπακετάρισμα των δεδομένων για τη μετέπειτα επεξεργασία αποτελεί ειδική ευθύνη του κάθε μέρους.

Client-side

Μία συνεδρία (στο εξής session) ξεκινάει πάντα από την πλευρά του client. Αρχικά, ο χρήστης προσκαλείται να “συνδεθεί” στον server εισάγοντας τα στοιχεία του server. Δεδομένου ότι ο χρήστης γνωρίζει και εισάγει σωστά τα στοιχεία, προχωράει στη βασική συνεδρία επισκέπτη (guest session).

Στη συνεδρία επισκέπτη, ο χρήστης καλείται επαναληπτικά να επιλέξει την επόμενη κίνησή του από τις εξής επιλογές (των οποίων η λειτουργία περιγράφεται αναλυτικά στην εκφώνηση της εργασίας):

- Σύνδεση (LogIn)
- Εγγραφή (SignIn)
- Έξοδος (Exit)

Κάθε μια από τις παραπάνω εντολές μεταβάλλει την τρέχουσα κατάσταση του client. Κάθε φορά που ο client μεταβαίνει σε μία νέα κατάσταση, στέλνει κατάλληλο μήνυμα-αίτημα (στο εξής request) στον server ώστε να βρίσκονται πάντα σε αντίστοιχες καταστάσεις και σε πλήρη συγχρονισμό.

Αν ο χρήστης μεταβεί στην κατάσταση εγγραφής, καλείται να προσδιορίσει το όνομα του υπό-κατασκευή λογαριασμού και τον κωδικό του. Κατόπιν, δημιουργείται και αποστέλλεται κατάλληλο request προς τον server, καθώς και τα απαραίτητα δεδομένα για να πραγματοποιήσει την εγγραφή από την πλευρά του (server-side).

Αν ο χρήστης μεταβεί στην κατάσταση σύνδεσης, καλείται να πληκτρολογήσει τα στοιχεία του λογαριασμού του. Και πάλι τα στοιχεία υποβάλλονται στον server για ταυτοποίηση και κατόπιν το έτοιμο σύνδεσης είτε γίνεται δεκτό, είτε απορρίπτεται.

Εάν το αίτημα σύνδεσης γίνει δεκτό, ο χρήστης μεταβαίνει στην βασική συνεδρία εγγεγραμμένου χρήστη (user session). Οι δυνατές επιλογές του (οι οποίες περιγράφονται πλήρως από την εκφώνηση της εργασίας) είναι πλέον διευρυμένες και περιλαμβάνουν τα εξής:

- Δημιουργία και αποστολή e-mail (NewEmail)
- Εμφάνιση αλληλογραφίας (ShowEmails)
- Ανάγνωση e-mail (ReadEmail)
- Διαφραφή e-mail (DeleteEmail)
- Αποσύνδεση (LogOut)
- Έξοδος (Exit)

Η λογική που επικρατεί στη τρέχουσα κατάσταση είναι αντίστοιχη με αυτή της συνεδρίας επισκέπτη.

Εάν ο χρήστης μεταβεί στη κατάσταση δημιουργίας και αποστολής e-mail, ο χρήστης καλείται να συντάξει ένα e-mail. Συγκεκριμένα, παρέχει το όνομα του παραλήπτη, το θέμα και το περιεχόμενο του e-mail. Ο client δημιουργεί στη συνέχεια ένα request αποστέλλοντας τα δεδομένα στον server. Ο server προσπαθεί να δημιουργήσει και να αποστείλει το e-mail. Είτε το πετύχει, είτε όχι, ενημερώνει τον client (και ο client ενημερώνει τον χρήστη) με σχετικό μήνυμα. Κάθε μια από τις παραπάνω εντολές αποτελεί και ένα μήνυμα προς τον server.

Αν ο χρήστης αιτηθεί την εμφάνιση της αλληλογραφίας του, ο server ενημερώνεται με κατάλληλο αίτημα και ετοιμάζει μία επισκόπηση της τρέχουσας κατάστασης της αλληλογραφίας του χρήστη και την αποστέλλει στον client. Σε αυτήν την επισκόπηση περιλαμβάνονται μεταξύ άλλων τα μοναδικά αναγνωριστικά του κάθε e-mail μαζί με μία σύντομη περιγραφή του e-mail (το θέμα).

Με παρόμοιο τρόπο λειτουργού οι καταστάσεις ανάγνωσης συγκεκριμένου e-mail και διαγραφής συγκεκριμένου e-mail. Ο χρήστης παρέχει το μοναδικό αναγνωριστικό του e-mail που τον ενδιαφέρει (θεωρείται ότι το γνωρίζει από προηγούμενη εμφάνιση της αλληλογραφίας) και ο server πραγματοποιεί την επιθυμητή λειτουργία.

Τελικά, σε κάθε περίπτωση, ο χρήστης μπορεί να μεταβεί στην κατάσταση εξόδου, κατά την οποία ο client αποχωρεί πρώτος από τη συνεδρία και ο server ενημερώνεται για να αποδεσμεύσει τους πόρους.

Server-side

Από τη πλευρά του server, υπάρχουν δύο αξιοσημείωτες οντότητες. Αυτή του ίδιου του server, και αυτή της συνεδρίας (session).

Ο server είναι υπεύθυνος για τη διαχείριση των δεδομένων – την τήρηση και την επεξεργασία τους – τη δημιουργία λογαριασμών, και τις πράξεις επί αυτών. Επιπλέον, είναι υπεύθυνος για την “ακοή” εισερχόμενων συνδέσεων και τον παραλληλισμό του μέσω νημάτων (threads).

Το session είναι υπεύθυνο για την υλοποίηση του πρωτοκόλλου επικοινωνίας μεταξύ του server και του client. Συγκεκριμένα, χρησιμοποιείται για τη παραγωγή και παραλαβή μηνυμάτων, την επεξεργασία τους και την αποστολή τους πίσω στον client.

Οι βοηθητικές οντότητες είναι αυτή του λογαριασμού (account) και του ομώνυμου e-mail. Καθεμία από αυτές αποτελεί μια απλή δομή δεδομένων που διαχειρίζεται αυτόνομα τα περιεχόμενά της και παρέχει την κατάλληλη αφαίρεση στα ανώτερα επίπεδα οντοτήτων. Οι λειτουργίες τους περιγράφεται πλήρως στην εκφώνηση της εργασίας.

Επεξήγηση Κώδικα

Η Κλάση Email

Πρόκειται για τη πιο θεμελιώδη οντότητα του συστήματος. Είναι υπεύθυνη για την ομαδοποίηση των επιμέρους στοιχείων ενός e-mail. Παρέχει στοιχειώδεις μεθόδους πρόσβασης (στο εξής setters και getters), και τη μέθοδο makeSeen η οποία ορίζει ρητά ότι το τρέχον e-mail έχει προβληθεί. Επιπλέον, υπερκαλύπτεται η μέθοδος toString, ώστε να μπορούν οι πελάτες της κλάσης να έχουν μία γενική εικόνα του αντικειμένου σε ένα string.

Η Κλάση Account

Πρόκειται για βασική οντότητα του συστήματος και μπορεί να λογιστεί ως δομή δεδομένων. Συγκεκριμένα, ομαδοποιεί τις απαραίτητες λεπτομέρειες ενός λογαριασμού εγγεγραμμένου χρήστη, και διατηρεί την αλληλογραφία του (mailbox). Επίσης, παρέχει τους κατάλληλους setters και getters, ορίζει λειτουργίες επί της αλληλογραφίας και παρέχει μία μέθοδο αναπαράστασης της παρούσας κατάστασή αυτής.

Η μέθοδος submitEmail παραδίδει ένα εισερχόμενο e-mail στο τρέχοντα λογαριασμό, ενώ η μέθοδος deleteEmail διαγράφει το επιθυμητό e-mail από την αλληλογραφία. Επιπλέον, σημαντικός είναι ο ρόλος της representMailbox καθώς επιστρέφει στον πελάτη της κλάσης τα υπάρχοντα e-mails μαζί με το μοναδικό τους αναγνωριστικό με τη μορφή string, ώστε ο χρήστης να μπορεί αργότερα να προσδιορίσει πιο e-mail επιθυμεί να επεξεργαστεί.

Η Κλάση Session

Πρόκειται για την υλοποίηση του πρωτοκόλλου από την πλευρά του server. Δημιουργεί διακριτά αντικείμενα-νήματα κατά την εκτέλεση του προγράμματος επιτρέποντας την παράλληλη σύνδεση πολλών χρηστών. Διατηρεί στατικά το ολικό σύνολο των sessions που έχουν παραχθεί και αναθέτει στο κάθε session το μοναδικό του αναγνωριστικό. Ακόμα, υπερκαλύπτει τη μέθοδο run η οποία είναι υπεύθυνη για την επικοινωνία με τον client και τον server και εκτελεί τα αιτήματα του πρώτου μέσω του τελευταίου.

Η Κλάση MailServer

Πρόκειται για τον κεντρικό επεξεργαστή των αιτημάτων του χρήστη προς τον λογαριασμό του. Είναι υπεύθυνη για την αποδοχή συνδέσεων των clients, την ανάθεση τους σε διακριτά sessions, και τη γενική λειτουργία σε μόνιμη βάση. Επιπλέον, υλοποιεί τις ζητούμενες λειτουργίες register (δημιουργία νέου λογαριασμού), login (ταυτοποίηση χρήστη), newEmail (δημιουργία και αποστολή νέου e-mail), readEmail (ανάγνωση υπάρχοντος e-mail), deleteEmail (διαγραφή υπάρχοντος e-mail) και showEmails (επισκόπηση αλληλογραφίας χρήστη), των οποίων οι λειτουργίες ορίζονται αναλυτικά στην εκφώνηση της εργασίας.

Για τους λόγους πληρότητας, παρέχεται επιπλέον η μέθοδος populateAccounts η οποία δημιουργεί ορισμένους εικονικούς χρήστες, και αποστέλλει ορισμένα e-mails εκ μέρους αυτών και προς αυτούς. Οι λογαριασμοί είναι “ανοιχτοί”, κατά τα ζητούμενα, και οι κωδικοί τους μπορούν να βρεθούν στο αντίστοιχο αρχείο.

Η Κλάση MailClient

Πρόκειται για τη συγκεντρωτική κλάση που εκτελεί τον ρόλο του client και της διεπαφής χρήστη. Είναι υπεύθυνη για την υλοποίηση του πρωτοκόλλου επικοινωνίας με τον server, καθώς και με τον τελικό χρήστη.

Αρχικά, παρέχονται ορισμένες βοηθητικές μέθοδοι για την απλοποίηση του επαναλαμβανόμενου κώδικα. Συγκεκριμένα, η μέθοδος print εκτυπώνει απλώς το επιθυμητό μήνυμα σε μία νέα γραμμή στο τερματικό του χρήστη. Η μέθοδος popup προσομοιώνει την ανάδυση κάποιου ειδικού “παραθύρου” εκτυπώνοντας το επιθυμητό μήνυμα πλαισιώνοντάς το με ειδικούς χαρακτήρες. Αντίστοιχα, η μέθοδος read αναμένει την είσοδο του χρήστη από το πληκτρολόγιο εκτυπώνοντας κάποιο προτρεπτικό σύμβολο (prompt). Επιπρόσθετα, οι μέθοδοι printGuestSession και printUserSession εκτυπώνουν τις τρέχουσες δυνατές επιλογές του χρήστη στο τερματικό του.

Περίληπτικά, η σύνδεση του χρήστη περιλαμβάνει τρεις βασικές καταστάσεις – τη συνεδρία επισκέπτη, τη συνεδρία χρήστη και τον τερματισμό της συνεδρίας. Καθεμιά από αυτές τις καταστάσεις υλοποιείται σε ξεχωριστή μέθοδο (αντίστοιχα guestSession, userSession και exit) και είναι υπεύθυνη για την αποδοχή αιτημάτων από τον χρήστη ή τον server, την επεξεργασία τους και, ενδεχομένως, την εκτύπωσή τους στον χρήστη ή την αποστολή τους στον server.

Επεξήγηση Χρήσης

Για τη λειτουργία του προγράμματος απαιτούνται τα αρχεία “MailServer.jar” και “Client.jar” που επισυνάπτονται υπό τον φάκελο “jar”. Επίσης, συνίσταται η χρήση της όγδοης τουλάχιστον έκδοσης της γλώσσας Java.

Αρχικά, χρειάζεται να αρχικοποιηθεί ο server. Αρκεί η ακόλουθη εντολή, σε κάποιο τερματικό που βρίσκεται στον ίδιο φάκελο με τα .jar αρχεία:

```
$ java -jar MailServer.jar
```

Το πρόγραμμα θα ζητήσει από τον διαχειριστή του συστήματος να εισάγει την επιθυμητή θύρα (port) στην οποία θα “ακούει” ο server για εισερχόμενες συνδέσεις. Εάν επιτευχθεί η σύνδεση του server με το δεδομένο port, εμφανίζεται κατάλληλο μήνυμα.

Κατόπιν, χρειάζεται να αρχικοποιηθεί κάποιος client. Αρκεί η ακόλουθη εντολή σε κάποιο άλλο τερματικό που βρίσκεται στον ίδιο φάκελο με τα .jar αρχεία:

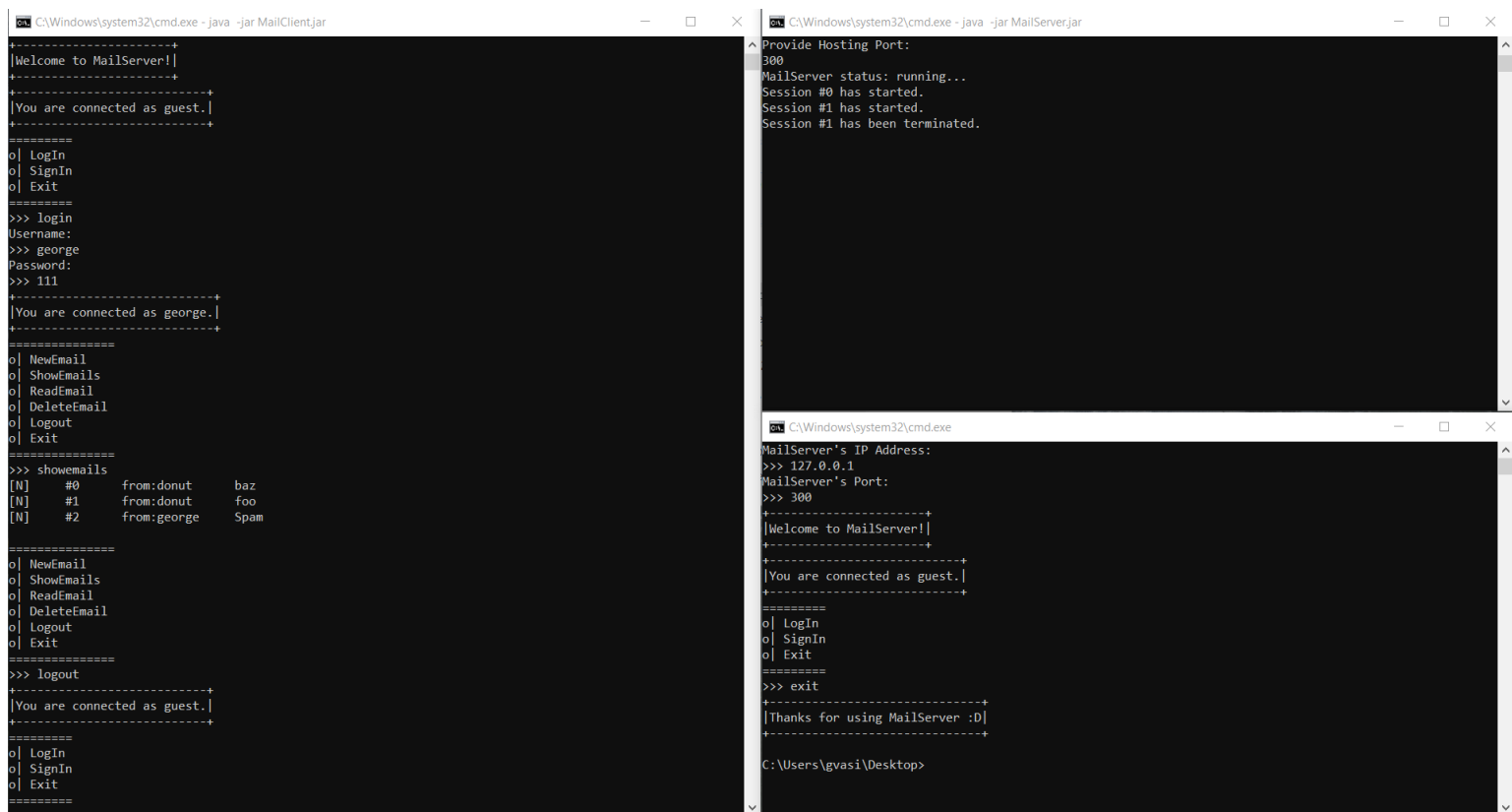
```
$ java -jar MailClient.jar
```

Το πρόγραμμα θα ζητήσει από τον χρήστη του προγράμματος να εισάγει την διεύθυνση IP του server, καθώς και το port που καθόρισε στο προηγούμενο βήμα. Εάν η σύνδεση επιτευχθεί εμφανίζεται κατάλληλο μήνυμα και το πρόγραμμα δουλεύει όπως αναμένεται.

Ας σημειωθεί ότι δεν περνιούνται παράμετροι σε κανένα από τα εκτελέσιμα αρχεία από το σύστημα (τερματικό). Η παραμετροποίηση γίνεται εσωτερικά των προγραμμάτων.

Εποπτικά Παραδείγματα Χρήσης

Εικόνα 1 – Δύο διαφορετικές συνδέσεις εκτελούν διαφορετικές εργασίες ανεξάρτητα και παράλληλα.



The image displays three terminal windows illustrating the parallel execution of MailServer and MailClient processes.

Top Left Window (MailClient): Shows the client interface. It starts with a welcome message, followed by a login attempt as 'george'. The user is then connected as 'george'. The user interacts with the menu, viewing a list of emails (from:donut, baz; from:donut, foo; from:george, Spam). The user then logs out and is connected as 'guest'.

Top Right Window (MailServer): Shows the server interface. It prompts for a hosting port (300) and displays the status: 'MailServer status: running...'. It also shows session information: 'Session #0 has started.', 'Session #1 has started.', and 'Session #1 has been terminated.'.

Bottom Window (MailServer): Shows the server interface. It displays the IP address (127.0.0.1) and the port (300). It then shows the client interface, including the welcome message, login attempt as 'george', and the list of emails. The user then logs out and is connected as 'guest'.

Εικόνα 2 – Τυπική συνομιλία πελάτη-εξυπηρετητή κατά τη σύνδεση ενός χρήστη.

