

תרגיל מעשי 1

תיאור פרוטוקול התקשורות:

סוג המשתנים בתוך הודעה:

- (1) **מספר**: תמיד יהיה בן ארבעה בתים בצורה של network byte order.
- (2) **מחרוזות**: רצף של בתים (כולל null בסופם), שקודם לכן יופיע מספר המייצג את אורך המחרוזת בביתים (כולל הnull בסוף המחרוזת).

רישא של הודעה:

כל הודעה מתחילה בשני מספרים בעל תפקיד קבוע:

- (1) **מספר המייצג את אורך ההודעה** העתידה לבוא. (כלומר, לא כולל את ארבעת הבתים המייצגים את אורך ההודעה).
- (2) **מספר המייצג את סוג ההודעה**.

| Byte: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... |
|-------|--------------|---|---|---|--------------|---|---|---|------|-----|
| Value | Message size | | | | Message Type | | | | Data | |

סוגי ההודעות ותיאורן:

1. Login request

תיאור:

מיד לאחר יצירת התקשרות עם השרת, הקליינט מבקש להתחבר עם שם משתמש וסיסמא.

מידע בהודעה (לפי הסדר):

- **מספר** ההודעה הינו 1.
- מכיל 2 **מחרוזות**:
 - שם משתמש.
 - סיסמת משתמש.

תשובה להודעה:

הודעת General message.

2. Show inbox

תיאור:

הקליינט מבקש מהשרת את רשימת ה-email-ים של המשתמש המחובר שנמצאים בשרת.

מידע בהודעה (לפי הסדר):

- **מספר** ההודעה הינו 2.

תשובה להודעה:

בהצלחה: הודעת Show inbox respond.
בכשלון: הודעת General Message מתאימה.

3. [Show inbox respond](#)

תיאור:

תגובה מהשרת לקליינט המכילה את רשימת ה-email-ים של המשתמש המחובר בשרת.

מידע בהודעה (לפי הסדר):

- **מספר** ההודעה הינו 3.
- מכיל **מחרוזות** משורשרות (מספר המחרוזות לא ידוע) כך שכל מחרוזת מכילה את הפרמטרים הבאים מופרדים ברווחים:
 - **מספר המייל בצד השרת.**
 - שולח ההודעה.
 - נושא ההודעה.

תשובה להודעה:

אין.

4. [Get mail](#)

תיאור:

הקליינט מבקש מהשרת את התוכן של הודעה ספציפית של המשתמש המחובר.

מידע בהודעה (לפי הסדר):

- **מספר** ההודעה הינו 4.
- מכיל **מספר** המייצג את **מספר המייל בצד השרת** (ראה Show inbox respond)

תשובה להודעה:

בהצלחה: הודעת Get mail respond.
בכשלון: הודעת General Message מתאימה.

5. [Get mail respond](#)

תיאור:

השרת מחזיר לקליינט תוכן של ההודעה הספציפית שהוא ביקש.

מידע בהודעה (לפי הסדר):

- **מספר** ההודעה הינו 5.
- מכיל 4 **מחרוזות**:
 - שם המשתמש ממנו נשלח המייל.
 - שמות המשתמשים אליהם נשלח המייל (מופרדים בפסיקים).
 - נושא המייל.
 - תוכן המייל.

תשובה להודעה:

אין.

Delete mail 6.

תיאור:

הקליינט מבקש מהשרת למחוק הודעה ספציפית מתיבת ההודעות של המשתמש המחובר בשרת.

מידע בהודעה (לפי הסדר):

- מספר ההודעה הינו 6.
- מכיל מספר המייצג את מספר המייל בצד השרת (ראה [Show inbox respond](#))

תשובה להודעה:

הודעת General message מתאימה.

Compose mail 7.

תיאור:

הקליינט מבקש מהשרת להוסיף הודעה לכל התיבות האישיות של הנמענים בשם המשתמש המחובר בשרת.

מידע בהודעה (לפי הסדר):

- מספר ההודעה הינו 7.
- מכיל 3 מחרוזות:
 - שמות המשתמשים אליהם ישלח המייל (מופרדים בפסיקים).
 - נושא המייל.
 - תוכן המייל.

תשובה להודעה:

הודעת General message.

General message 8.

תיאור:

הודעה שהשרת יכול להחזיר לקליינט בכל שלב בזמן החיבור, כתגובה לבקשה של הקליינט.

מידע בהודעה (לפי הסדר):

- מספר ההודעה הינו 8.
- מכיל מספר המייצג את סטטוס הפעולה האחרונה שהשרת ביצע:
 - (0) הפעולה עברה בהצלחה.
 - (1) תשובה לפעולת Login: שם המשתמש לא ידוע.
 - (2) תשובה לפעולה Login: הסיסמא לא נכונה.
 - (3) המשתמש אינו מחובר ולכן אינו יכול לבצע פעולה.
 - (4) תשובה לפעולות Get/Delete mail: מספר המייל לא ידוע.
 - (5) תשובה לפעולת Login: המשתמש כבר מחובר לשרת.
 - (10) שגיאה בצד השרת – נסה שוב לבצע את הבקשה.
 - (11) שגיאה קשה בצד השרת שבעקבותיה הוא מפסיק לעבוד.
 - (12) הבקשה לא חוקית – נסה שוב לבצע את הבקשה.
 - (13) בעיה בחיבור ולכן החיבור ינותק מצד השרת.

תשובה להודעה:
אין.

9. [Welcome](#)

תיאור:

ההודעה הראשונה שנשלחת: מהשרת אל הלקוח עוד לפני ביצוע Login

מידע בהודעה (לפי הסדר):

- מספר ההודעה הינו 9.
- מכיל **מחרוזת** שמייצגת הודעת ברכה למשתמש שהתחבר לשרת.

תשובה להודעה:
אין.

10. [Quit](#)

תיאור:

ההודעה האחרונה שנשלחת: הלקוח מודיע לשרת שהוא מסיים את ההתקשרות בינהם.

מידע בהודעה (לפי הסדר):

- מספר ההודעה הינו 10

תשובה להודעה:
אין.

הרצת התוכנית:

כמתואר בדרישות התרגיל.

שרת:

`./mail_server users_file [port]`

לקוח:

`./mail_client [hostname [port]]`

במידה ושדות אופציונאליים לא נכתבו, ערכי ברירת המחדל הם:

פורט: 6423

שם מארח: localhost

מבנה הקוד:

תשתית:

מחלקות המשרתות את צד השרת והלקוח.

Common - משותף

מכילה מידע כללי משותף גם לשרת וגם ללקוח, כגון: קידוד של ההודעות העוברות ביניהם, פונקציות הדפסה למסך ועוד.

Packet – משותף

מחלקה המשמשת לקריאה וכתיבה של מידע בצורה נוחה.

היא עוטפת כתיבה וקריאה של מחרוזות ומספרים בצורה המתואמת ל[פרוטוקול שרת-לקוח שלנו](#).

Socket - משותף

מחלקה העוטפת socket. מעבר למעטפה הרגילה של פונקציות הAPI, מכילה גם פונקציות כתיבה וקריאה (sendMessage, recvMessage) המשתמשות ב-[Packet](#) ומודעות ל[פרוטוקול שרת-לקוח שלנו](#).

FDSet – צד שרת

מחלקה שעוטפת את המבנה איתו עובדים בשימוש של פונקצית select.

User – צד שרת

מחלקה המייצגת משתמש.

MailObj – צד שרת

מחלקה המייצגת אימייל.

Inbox - צד שרת

מחלקה המייצגת תיבת דואר של משתמש. מכילה רשימת המיילים שנשלחו למשתמש מסויים, (רשימה של [MailObj](#)), ומידע אודות המשתמש המשויך אליה (אובייקט [User](#)).

ServerSessionSocket – צד שרת

מחלקה שמציגת session שפעיל כרגע בין הלקוח ובין השרת.

יורשת מ-Socket ומכילה בתוכה [Inbox](#).

Client

מחלקה המייצגת את הלקוח. מבצעת את תהליך התחברות לשרת, קבלת פקודות מהמשתמש, שליחתם לשרת ודיווח על תוצאות הפקודות.

Server

מחלקה המייצגת את השרת. מבצעת את תהליך האתחול שלו, מאזינה לבקשות ומבצעת אותם.

מקרים שלא הוגדרו

צד לקוח

• טיפול בשגיאות:

- כאשר יש שגיאה (מכל סוג) עד שלב Login (כולל) אין למשתמש הזדמנות נוספות להכניס נתונים והתוכנית מפסיקה.
- לאחר ההתחברות ישנם סיבות שונות שהתוכנית יכולה להפסיק בגינם, כגון:
 - הודעת שגיאה חמורה מצד השרת שבעקבותיו הוא צריך לסגור את החיבור (ראה [General Message](#), הודעות 11 ו-13)
 - בעיות בsocket של הלקוח.
 - בעיות הקצאת זיכרון בצד הלקוח.
 - בעיות בקבלת או שליחת מידע מ/לשרת.
 - בעיות בפרסור הודעה ברמת הפרוטוקול (לדוגמא, קבלת general message status לא ידוע)

במידה והשגיאה אינה חמורה: לדוגמא הכנסת פקודה לא קיימת כקלט, או פורמט לא נכון של קלט - תוכנית הלקוח תמשיך לעבוד ותאפשר למשתמש להכניס פקודות נוספות.
בכל מקרה של שגיאה (בכל שלב בריצת התוכנית) תכתב הודעה מתאימה למשתמש.

• קלטים מהמשתמש:

- כפי שמוגדר במסמך התרגיל הקלטים בפעולת Login ובפעולת Compose יכילו את ה prefix:
 - User:
 - Password:
 - To:
 - Subject:
 - Text:
- בסופם צריך להופיע רווח, ואחריו **המידע**.
כל הקלטים הנ"ל יכולים להכיל רווחים **בחלק המידע**.
- אין בדיקת נכונות של הערכים בשדה To: בפקודת Compose מצד הלקוח.

• העברת פרמטרים להרצת התוכנית:

- במידה ומועברים פרמטרים נוספים לתוכנית, אין שום התייחסות אליהם (התוכנית ממשיכה לרוץ כרגיל).
- התוכנית מניחה שהפרמטר השני (של הפורט) הוא מספר פורט חוקי.

- התוכנית תומכת בכתובות IPv4 בלבד.
- אורך מחרוזת מקסימלית בכל סוג הודעה הוא 10000 תווים.
- אורך שורה מקסימלית כקלט מהמשתמש הוא 2000 תווים.

צד שרת

- כאשר יש קבלה של הודעה לא חוקית, השרת עלול לנתק את התקשורת עם הקליינט ובכל מקרה ינסה לשלוח אליו הודעת General Message.
- קובץ משתמשים:
 - במידה והקובץ לא קיים, השרת לא יתחיל לעבוד.
 - במידה והקובץ קיים אך הוא ריק אז השרת יתחיל לעבוד אבל לא יהיו משתמשים שיוכלו להתחבר.
 - במידה והקובץ לא בפורמט הנכון, השרת יתחיל לעבוד אבל רשימת המשתמשים הטעונים אינה מוגדרת ויכול להכיל רק חלק מהמשתמשים או אף אחד מהם.
- פקודת Compose, טיפול בשדה To:
 - במידה ואחד המשתמשים (או יותר) לא קיים בשרת, השרת מתעלם ממנו ומטפל בשאר ההודעה כרגיל.
 - במידה ומשתמש מסוים קיים יותר מפעם אחת, ההודעה משוכפלת בתיבת המייל שלו כמספר ההופעות שלו.
 - שמות המשתמשים צריכים להיות מופרדים בפסיקים ללא רווחים.
- התוכנית מניחה שהפרמטר של הפורט הוא מספר פורט חוקי.
- אורך מחרוזת מקסימלית בכל סוג הודעה הוא 10000 תווים.