

## רשתות תקשורת – תרגיל מעשי מס' 1

אלמוג זלצמן / 204312763 / almogzeltsman@gmail.com

יהב אביגל / 200921740 / yahavigal@gmail.com

### הגדרות

:Headers

NW – הגדרת ה-structs והקבועים הלוקחים חלק בפקודות השונות

:Modules

Message – פונקציות המשמשות לקבלה ושליחה של מידע באמצעות sockets על בסיס הפרוטוקול שהוגדר.

Mail\_Server – מימוש פרוטוקול ההתקשרות והלוגיקה של שרת המייל

Client – מימוש פרוטוקול ההתקשרות והלוגיקה של הלקוח

:Structs

כללי

MSG – כל הודעה מורכבת מ-OPCODE (ENUM המתואר בסוף המסמך) המגדיר את סוג ההודעה (ומכאן נגזר לפי הפרוטוקול גם אופן הטיפול באותה ההודעה), אורך ההודעה וגוף ההודעה אשר מוגבל לאורך מקסימלי של MAXSIZE=65535 תווים. האובייקט כאמור הינו בפורמט הבא:

```
Struct {
    OPCODE      opcode;
    short       length;
    char        msg[MAXSIZE];
}
```

MAIL – כל מייל מורכב משולח (באורך 50 תווים לכל היותר), קבוצת נמענים (לכל היותר 20, כל אחד באורך לכל היותר 50 תווים), נושא (באורך 100 תווים לכל היותר) ותוכן (באורך 2000 תווים לכל היותר).

משתמשים

USER – לכל משתמש מוגדר שם משתמש (באורך לכל היותר 50 תווים) וכמו כן סיסמא (באורך לכל היותר 50 תווים).

USERLIST – רשימת המשתמשים בשירות המייל מכילה רשימה של USER-ים (לכל היותר 20 משתמשים), inbox להודעות של כל המשתמשים (לכל היותר 32000 הודעות לכל משתמש), מספר ההודעות שקיבל כל משתמש ומערך דגלים isMail המשמש למחיקת הודעות יעילה.

**הערה:** האובייקט הנ"ל מוגדר רק פעם אחת בהרצת התוכנית, כאשר Mail\_Server מאותחל. יש לציין כי לשם ייתור השימוש בהקצאות דינאמיות בפרויקט, כפי שהונחנו ע"י המתרגל, האובייקט עושה שימוש במספרים אשר הוגדרו כקבועים בהוראות התוכנית. האובייקט אם כך שומר מראש מיקום לכל מייל פוטנציאלי עבור כל משתמש פוטנציאלי (ובכל מייל מיקום לכל האורך הפוטנציאלי שלו). מסיבות אלה האובייקט הינו גדול במיוחד – כ-2GB, וככזה, נמצא שעלול להכשיל קריאה פוטנציאלית של valgrind עם Mail\_server [עקב באגים פנימיים של valgrind]. נבקש אם כן את התחשבות הבודק בנושא זה.

ACTIVEUSER – מחזיק את המשתמש המחובר לשרת בזמן אמת.

## פרוטוקול התקשורת

ראשית נציין:

- בכל שגיאת תקשורת מכל סוג בעת קבלת או שליחת הודעה, תודפס הודעת שגיאה למסך כך שהמשתמש ידע איזה שלב נכשל, והתוכנית תיכשל ותיסגר.
- לכל הודעה שהשרת יקבל, תחילה יבדוק את ה-OPCODE שלה, ויטפל בהתאם אליו, כלומר הטיפול הראשוני מסווג לפי ה-OPCODE.
- השרת משרת לקוח אחד בו זמנית.

להלן תיאור התהליכים האפשריים במערכת:

- **LOGIN** – עם הקמת ה-mail\_server הוא קורא קובץ המכיל שמות משתמש וסיסמאות, ובעזרת קובץ זה נבנית רשימת המשתמשים. הקובץ, כמתואר בהוראות הפרויקט הוא בפורמט הבא:

```
Username1    password1
Username2    password2
```

כאשר בין השם משתמש והסיסמא מפריד טאב אחד בדיוק ('/t').

התוכנה מניחה כי הקובץ **קיים**, ובפורמט **תקין**.

כאשר לקוח חדש רוצה להתחבר, אם נוצר חיבור הוא יקבל הודעת WELCOME, הודעת התחברות ראשונית מהשרת אשר תוכנה יוצג למסך המשתמש. אם החיבור נכשל, תודפס על המסך הודעה באשר לתהליך הספציפי אשר נכשל (למשל, יצירת ה-socket).

בשלב זה הלקוח מחכה לקלט מהמשתמש (שם המשתמש והסיסמא שלו) אשר תישלח לשרת, בפורמט הבא:

User: username1

Password: password1

ההודעה הנשלחת לשרת תהיה בפורמט הבא:

OPCODE	Length	Msg
LOGIN	Length of msg	Username+\0+password+\0

ואם הוא נמצא ברשימת המשתמשים ייוצר החיבור ותישלח הודעה "connected to server".

ההודעה אשר חוזרת מהשרת במקרה הזה היא בפורמט:

OPCODE	Length	Msg
LOGIN_SUCCESS	0	\0

אם ההודעה נשלחה בפורמט לא תקין, התוכנה תניח כי המשתמש ניסה להקליד הוראה אחרת שאינה התחברות, ותדפיס את הודעת השגיאה:

You are not logged in. Please log in by writing exactly:

User: username

Password: password

נציין שפקודת QUIT תעבוד בשלב הזה (במקום כתיבת שם המשתמש בלבד, לא במקום כתיבת הסיסמא אם שם המשתמש נכתב כבר), ותצא מהתוכנית של ה-client כרגיל (פורמט ההודעה בהסבר על פקודת ה-QUIT).

אם שורת שם המשתמש נשלחה בפורמט תקין אך שורת הסיסמא לא נשלחה בפורמט תקין, תודפס הודעת השגיאה:

Error while getting password

Please log in by writing exactly:

User: username

Password: password

אם ההודעה נשלחה בפורמט תקין אך המשתמש לא נמצא או שהמשתמש נמצא והסיסמא לא הייתה תקינה, תודפס למסך ההודעה:

Username or password incorrect

ההודעה אשר חוזרת מהשרת במקרה הזה היא בפורמט:

OPCODE	Length	Msg
LOGIN_FAIL	0	\0

בכל המצבים האלה (מלבד התחברות תקינה או שגיאת תקשורת), תינתן ללקוח אפשרות לנסות שנית את כל התהליך.

- SHOW\_INBOX – זוהי פקודה הנשלחת לשרת, בעקבותיה תוצג רשימת המיילים הקיימים במערכת ושייכים למשתמש המחובר. ההודעה אשר נשלחת לשרת במקרה הזה היא בפורמט:

OPCODE	Length	Msg
SHOW_INBOX	0	\0

בתגובה השרת שולח ללקוח הודעה אשר אם היא בפורמט הבא –

OPCODE	Length	Msg
SHOW_INBOX	0	\0

הלקוח מבין שתיבת המייל של המשתמש בשרת ריקה ולא יודפס שום תוכן למשתמש. אם ההודעה היא בפורמט הבא –

OPCODE	Length	Msg
SHOW_INBOX	Length(String(number))	number

ההודעה מתארת את מספר ההודעות שב-inbox של המשתמש. לאחר מכן כמספר הפעמים המופיע ב-number, הלקוח יחכה להודעות נפרדות מהשרת עבור כל מייל אשר נמצא ב-Inbox. פורמט ההודעה במקרה הזה יהא:

OPCODE	Length	Msg
SHOW_INBOX	Length(msg)	MAIL_ID FROM "SUBJECT"\n

כך שהתוכן שלה מכיל את מספר ההודעה (mail\_ID), ממי ההודעה נשלחה ואת נושא ההודעה. לאחר קבלת ההודעה מהשרת, הלקוח מדפיס לצג המשתמש את ההודעות בפורמט הנ"ל.

- GET\_MAIL – פקודה הנשלחת מהלקוח לשרת, בה הלקוח מציין בתוכן ההודעה את מזהה המייל אותו ברצונו לקבל חזרה. פורמט ההודעה במקרה הזה יהא:

OPCODE	Length	Msg
GET_MAIL	Length(msg)	MAIL_ID

השרת מחלץ את המייל המבוקש מתוך ה-inbox, מעתיק את תוכנו למייל חדש ושולח את העותק חזרה ללקוח. פורמט ההודעה הנשלחת מהשרת יהא:

OPCODE	Length	Msg
GET_MAIL	Sizeof(MAIL)	(char*) mail

במידה ומזהה המייל לא חוקי, או שההודעה לא קיימת במאגר (מחוקה או לא נשמרה מעולם) הלקוח יקבל בחזרה הודעה בפורמט הבא:

OPCODE	Length	Msg
INVALID	0	\0

יודפיס למשתמש שהמזהה INVALID. אחרת, יודפסו לצג המשתמש המוען, הנמענים, נושא ההודעה ותוכנה, כאשר הם מופרדים זה מזה באמצעות \n כמתואר בהוראות התרגיל.

- **COMPOSE** – בפקודה זו הלקוח מקבל כקלט מהמשתמש את פרטי הנמענים אליהם הוא שולח את ההודעה, ואת תוכן ההודעה. לאחר שליחת הפקודה **COMPOSE**, התוכנה מצפה מהמשתמש לקבל את ההודעה בפורמט המתואר בהוראות התרגיל במדויק.

1. אם המשתמש לא כתב "To:" בתחילת השורה הראשונה אז תודפס הודעת השגיאה הבאה:

```
Invalid command. Use exactly:
COMPOSE
To: user1,user2,...
Subject: subject...
Text: content...
```

2. אם המשתמש כתב "To:" אבל לא כתב שום נמענים לאחר מכן, תודפס הודעת השגיאה הבאה:

Message must have recipients

3. אם המשתמש לא כתב "Subject:" כמו שצריך, תודפס למסך הודעת השגיאה הבאה:

Error while getting subject

4. אם המשתמש לא כתב "Text:" כמו שצריך, תודפס למסך הודעת השגיאה הבאה:

Error while getting text

5. אם המשתמש הקליד שמות משתמשים אשר לא קיימים, תודפס למסך ההודעה הבאה:

One or more from the recipients don't exist, composing failed

בכל המקרים לעיל שליחת ההודעה תיכשל, ותינתן למשתמש הזדמנות נוספת להקליד **COMPOSE** ולהתחיל מהתחלה.

הלקוח מרכיב בעזרת נתונים אלה ובעזרת פרטי המשתמש מייל חדש (כ-struct) ושולח את המייל הזה בהודעה בפורמט הבא:

OPCODE	Length	Msg
COMPOSE	Sizeof(MAIL)	(char*) mail

כאשר mail מכיל בתוכו את השדות עם כל התוכן שהתקבל מהמשתמש, כפי שמתואר בתיאור ה-struct.

כאשר השרת מקבל הודעה זו, הוא מעתיק את תוכן ההודעה (המייל) לתוך מיילים חדשים כמספר הנמענים, ושומר כל עותק בתיבה של הנמען הרלוונטי. לאחר מכן גדל ה-inbox size של כל נמען רלוונטי ב-1. במידה והכל תקין השרת מחזיר ללקוח הודעה שמאשרת שהמיילים נשלחו, בפורמט הבא:

OPCODE	Length	Msg
COMPOSE	0	\0

אם אחד או יותר מהנמענים לא קיים, תישלח הודעה בפורמט הבא:

OPCODE	Length	Msg
INVALID	0	\0

- **DELETE\_MAIL** – פקודה הנשלחת מהלקוח לשרת, בה הלקוח מציין בתוכן ההודעה את מזהה המייל שברצונו למחוק. פורמט ההודעה במקרה הזה יהא:

OPCODE	Length	Msg
DELETE_MAIL	Length(msg)	MAIL_ID

השרת מכבה את הדגל isMail הרלוונטי לאותה הודעה, ומאותו הרגע היא נחשבת כמחוקה ולא ניתן יהיה לגשת אליה. אם מזהה המייל לא חוקי, הלקוח יקבל בחזרה הודעה בפורמט הבא:

OPCODE	Length	Msg
INVALID	0	\0

וידפיס למשתמש שהמזהה INVALID.



במידה והמזהה חוקי (בטווח שבין 0 לבין מספר ההודעות שבתיבה) אך ההודעה הייתה מחוקה מלכתחילה, המצב לא ישתנה והיא תישאר מחוקה, ולא תודפס הודעה מיוחדת למשתמש, תישלח במקרה הזה ההודעה הבאה מהשרת למשתמש:

OPCODE	Length	Msg
DELETE_MAIL	0	\0

- QUIT – פקודה הנשלחת מהלקוח לשרת. במידה והלקוח מעוניין להתנתק, הוא שולח לשרת הודעה בפורמט הבא:

OPCODE	Length	Msg
QUIT	0	\0

הוא מאפס את המשתנה השומר את המשתמש הנוכחי, וסוגר את ה-SOCKET עם המשתמש. נציין כי QUIT הינה פקודה של הלקוח בלבד, ולשרת אין פקודה כזו – לאחר שאותחל בהצלחה הוא אינו מסיים את ריצתו עד אשר התוכנית נסגרת ע"י המשתמש שהפעיל אותה.

## נספח 1 – תיאור ה-OPCODES

ה-OPCODE הינו ENUM אשר ערכיו מתוארים בטבלה שלהלן:

OPCODE	Value
INVALID	0
WELCOME	1
LOGIN	2
LOGIN_SUCCESS	3
LOGIN_FAIL	4
SHOW_INBOX	5
GET_MAIL	6
DELETE_MAIL	7
COMPOSE	8
QUIT	9