**Communication Networks**

**Practical Exercise 2**

## Name 1: Avi Caciularu

## ID 1: 203056585

## Email 1: [avi.c33@gmail.com](mailto:avi.c33@gmail.com)

## Name 2 Noam Gottlieb

## ID 2: 201606951

## Email 2: [noam.got@gmail.com](mailto:noam.got@gmail.com)

## Name 3: Mattan Serry

## ID 3: 304957673

## Email 3: [matanse@gmail.com](mailto:matanse@gmail.com)

**מכיוון שתרגיל זה הינו הרחבה של התרגיל הראשון, פרוטוקול ההתקשרות ומבנה התוכנית דומים בעיקרם. במסמך זה טקסט בעל פונט מודגש הוא סימן לשינוי או תוספת שהוכנסו לתרגיל הנוכחי. כמו כן השארנו תיאורים שהיו בתרגיל הראשון והוסרו בתרגיל השני, מסומנים ~~בקו חוצה~~.**

# מבנה התוכניות

תוכנת הלקוח רצה לפי הקובץ mail\_client.c ואילו תוכנת השרת רצה לפי mail\_server.c. שתי התוכניות נעזרות בספריה mail\_common.h/c שכתבנו שמכילה את כל הספריות (#include), קבועים (#define) והפונקציות המשותפות לשני הצדדים.

# תיאור הפרוטוקול

בתחילת התקשורת בין השרת ללקוח, לאחר שנוצר סוקט ביניהם, השרת שולח ללקוח בייט אופקוד LOG\_REQUEST (יוסבר בהמשך).

* הבהרת מונח: דאטה (Data; ברבים נכתוב "דאטות") היא רצף בתים באורך משתנה ובעלי משמעות מסויימת הנשלחת בסוקט. במידה ואנו מעוניינים לשלוח אינפורמציה כלשהי X, ראשית נשלח שני בתים המציינים את גודל האינפורמציה (שגודלה לא יעלה על 216 בתים) ומיד לאחר מכן נשלח את X. אם הצד השני מכוון לקליטת דאטה, ראשית הוא יקרא שני בתים כדי לפענח את |X|, ומיד לאחר מכן יקרא את X.לכן הגודל הריאלי של דאטה יהיה תמיד |X|+2 בתים. הפונקציות המשמשות הן sendData ו-recvData.
* הבהרת מונח: פרינט (Print) הוא רצף בתים באורך משתנה אשר נשלח מהשרת ללקוח ומטרתו להציג ללקוח מחרוזת טקסט כלשהי. הפרינט מורכב מבית אחד המייצג את האופקוד OP\_PRINT, ולאחר מכן דאטה המכילה את המחרוזת הטקסטואלית. כאשר הלקוח קורא את האופקוד OP\_PRINT, הוא יידע כי את הדאטה הבאה יש להציג למשתמש ע"י printf. לכן הגודל הריאלי של פרינט המכיל את המחרוזת S יהיה תמיד |S|+1 בתים (גודל האופקוד ועוד גודל הדאטה). הפונקציה המשמשת את השרת לשליחת הפרינט היא sendPrintToClient.
* הבהרת מונח: האלט (Halt) הוא קבלה או שליחה (בהתאם להקשר) של האופקוד OP\_HALT, שתפקידו יוסבר בהמשך.

נציג בעמודים הבאים את התנהגות כל אחד מהצדדים לאורך ההתקשרות.

# צד הלקוח

לאחר שהלקוח הקים בהצלחה את החיבור עם השרת, הוא נמצא בהאזנה מתמדת לבייט מהשרת – כלומר, רץ בלולאה אינסופית ומנסה לקרוא בייט אחד (אלא אם התקבל אופקוד OP\_HALT – יוסבר בהמשך). בייט זה הוא האופקוד אשר יקבע את הפעולה הבאה שעל הלקוח לבצע.

נסקור את האופקודים שהלקוח יכול לקבל מהשרת:

## LOG\_REQUEST

זהו מצב ייחודי לבקשת שם משתמש וסיסמה. במצב זה תוכנת הלקוח תבקש מהמשתמש להזין קלט בפורמט הבא:

User: username

Password: password

לאחר שהוזן קלט תקין, **יישלח לשרת אופקוד זהה ואחריו** תישלח דאטה המכילה את שם המשתמש ואחריה דאטה שמכילה את הסיסמה. לאחר מכן הלקוח יחזור למצב האזנה ולאחר מכן יישלח פרינט מהשרת אשר יודיע ללקוח אם ההתחברות הצליחה או כשלה. לאחר מכן יישלח אחד מהאופקודים הבאים:

* OP\_HALT – במידה וההתחברות הצליחה, השליטה חוזרת ללקוח וכל האפשרויות פתוחות בפניו (הסבר בהמשך)
* ~~LOG\_REQUEST – במידה והלקוח הזין שם משתמש וסיסמה שגויים, השרת ישלח אופקוד זה ע"מ לאפשר לו נסיונות התחברות נוספים. המספר המקסימלי של נסיונות אלו מוגדר ע"י הקבוע (#define) MAX\_LOGIN\_ATTEMPTS כאשר אצלנו הוא מוגדר כ-3. כאן גם המקום לציין כי במידה והלקוח הזין את הקלט בפורמט שגוי (למשל: כתב Username: <username>, לא עשה רווח אחרי המילה הראשונה וכד'), תוכנת הלקוח מאפשרת לו להזין שוב ושוב (ללא הגבלה) עד שיוזן קלט בפורמט תקין.~~ **לצורך פשטות, הורדנו בתרגיל הנוכחי את הפיצ'ר הזה – למשתמש ניתנת הזדמנות התחברות יחידה.**
* LOG\_KILL – ניתוק של המשתמש - ראו למטה.

## LOG\_KILL

במידה והלקוח **הזין שם משתמש וסיסמה לא נכונים** ~~הגיע למספר הנסיונות המקסימליים להתחברות,~~ יישלח אופקוד זה שיגרום לסגירת תוכנת הלקוח (יחד עם הדפסת הודעה מתאימה).

## OP\_PRINT

במצב זה תוכנת הלקוח קוראת דאטה אחת ומדפיסה אותה לטרמינל כמחרוזת טקסטואלית.

## OP\_HALT

במצב זה תוכנת הלקוח עוצרת את הקריאה מהשרת. זהו אופקוד המציין כי השרת סיים לשלוח את כל המידע שהלקוח מעוניין בו, והוא כעת חוזר ~~למצב האזנה ללקוח~~ **לטפל בלקוחות נוספים (אם יש כאלו)**. הלקוח מצידו, כעת חופשי להקליד ולבקש פעולות חדשות מהשרת. **תוכנת הלקוח מאתחלת קבוצת file descriptors שמכילה את STDIN וכן את הסוקט דרכו הוא מתקשר את הלקוח, ומתבצעת על קבוצה זו קריאת select. במצב תקין, כאשר הקריאה חוזרת ייתכנו אחד מ-2 תרחישים:**

* **במידה והמשתמש סיים להזין פקודה לביצוע הקריאה מופרעת ע"י STDIN והלקוח יוכל לסיים לבצע את הזנת המידע הנוסף הדרוש (רלוונטי בעיקר לפקודת COMPOSE). תוכנת הלקוח מתרגמת את הבקשה לאופקוד מתאים, אותו תשלח לשרת (ראו פירוט בצד השרת). יש לציין כי ע"מ שהשרת לא ימתין ללקוח מסוים, צד הלקוח מכין את כל המידע שהוא עומד לשלוח לשרת (ומוודא שהוא בפורמט תקין), ורק כשכל המידע מוכן הלקוח שולח לשרת את האופקוד ומיד לאחר מכן את המידע הרלוונטי.** נציין כי במידה והוזנה פקודה לא מוכרת, תוכנת הלקוח תדפיס הודעה מתאימה ותבקש הזנה חוזרת (ללא הגבלת מספר הנסיונות).
* **במידה ומשתמש לא סיים להזין פקודה והקריאה חוזרת – זה סימן שהגיעה הודעת צ'אט למשתמש – התוכנה תחזור לתחילת לולאת הפעילות של הלקוח, תקבל פרינט ותדפיס את ההודעה שהתקבלה, ולאחר מכן יתקבל האלט נוסף והמשתמש יוכל להמשיך מהמקום שבו נעצר.**

# צד השרת

**לפני שהשרת נכנס ללולאת הקוד המרכזית שבה הוא מגיב לבקשות מלקוחות, הוא מאתחל את קבוצת ה-file descriptors כך שתכיל בשלב ראשון את הסוקט שדרכו מתקבלות בקשות התחברות לשרת (קבוצה זו מעודכנת כאשר מתקבלים חיבורים חדשים או כאשר לקוחות מתנתקים). בכל איטרציה חדשה של הלולאה השרת מבצע קריאת select על קבוצת ה-FD ומחכה שתגיע בקשת התחברות (connect) או שליחה (של אופקוד) מלקוח. כשבקשה כזו מגיעה השרת רץ בלולאה נוספת על מספרי ה-FD עד fdmax (מס' ה-FD המקסימלי של סוקט של לקוח פעיל) וכשהוא מוצא את ה-FD ממנו הגיעה הבקשה הוא מטפל בה בהתאם.**

**ככלל, טיפול בבקשה ספציפית של לקוח מתבצע תוך רצף ידוע מראש של שליחות וקבלות בין השרת ללקוח, כלומר, מרגע שהתקבלה בקשה מלקוח השרת לא מבצע קריאות שנחסמות (כל קריאה חוסמת, כמו send למשל, תיענה לפי זרימת תוכנית הלקוח בקריאת recv מיידית, כך שלא תיחסם).**

**שינוי נוסף שכדאי לציין הוא שבתרגיל בעת קבלת בקשה מלקוח הזיהוי של המשתמש המתאים נעשה לפי ה-FD של הלקוח (אנו מניחים כי כאשר משתמש מתחבר למערכת הוא מתקשר בכל זמן החיים שלו דרך אותו FD). נכתבה לכך פונקציה ייעודית.**

~~אם השרת שלח אופקוד OP\_HALT, הוא בהאזנה מתמדת לבייט מהלקוח.~~ **לאחר ששרת ענה לבקשה שאינה בקשת התחברות הוא מקבל בייט בודד מהלקוח הרלוונטי.** בייט זה הוא האופקוד­­­ אשר יקבע את המשך ההתקשרות. ~~קבלת אופקודים מהלקוח אפשרית רק לאחר שלב האימות.~~

נסקור את האופקודים שהשרת יכול לקבל מהלקוח (אופקודים אלו נשלחים מהלקוח לאחר קבלת פקודה מתאימה כפי שהוגדר בתרגיל. לא קשה להבין איזה אופקוד נשלח לאחר איזו פקודה...):

## **LOG\_REQUEST**

**זהו האופקוד הראשון שצריך להתקבל מהלקוח לאחר שזה התחבר לשרת. אופקוד זה מסמן לשרת כי הלקוח רוצה להיכנס למערכת בעזרת שם משתמש וסיסמה, ולכן מיד לאחר קבלתו השרת מקבל מהלקוח את שם המשתמש בסיסמה ובודק האם הם מתאימים לרישומים הקיימים אצלו. במידה וכן נשלח פרינט עם הודעת התחברות ("Connected to server") ולאחר מכן האלט כך שהלקוח יכול לקבל הנחיות מהמשתמש. במידה והכניסה לא הצליחה, השרת שולח סיגנל LOG\_KILL שגורם לניתוק הלקוח (ראו הסבר גם בצד הלקוח).**

## OP\_SHOWINBOX

במצב זה תוכנת השרת סורקת את תיבת המייל של הלקוח ושולחת ללקוח פרינט עבור כל מייל שנמצא בתיבה, בפורמט הבא:

# sender "subject"

בסיום שליחת הפרינטים יישלח האלט ללקוח.

## **OP\_SHOW\_ONLINE\_USERS**

**כפי שניתן להבין, לאחר קבלת אופקוד זה השרת סורק את רשימת המשתמשים הפעילים (משתמש פעיל יזוהה לפי FD חיובי), ולאחר מכן שולח פרינט ללקוח עם רשימת המשתמשים המחוברים ברגע זה (כולל המשתמש המבקש), בפורמט הבא:**

**Online users: <user1>, <user2>, …, <user\_k>**

**בסיום שליחת הפרינט יישלח האלט ללקוח.**

## **OP\_CHAT\_MSG**

**אופקוד זה נשלח לשרת כאשר לקוח מעוניין לשלוח הודעת צ'אט ללקוח אחר. השרת מבצע לאחר קבלת האופקוד שתי קריאות של דאטה (recipient ו-content – נמען ותוכן).**

* **במידה והנמען לא זוהה נשלחת הודעה מתאימה לשולח ההודעה.**
* **במידה והנמען מחובר יישלח אליו פרינט בפורמט הבא:**

**New message from <sender>: <content>**

**כאשר <sender> הוא המשתמש ששלח את ההודעה.**

* **אם הנמען אינו מחובר יישלח אליו מייל בפורמט הבא:**

**From: <sender>**

**To: <recipient>**

**Subject: Message received offline**

**Text: <content>**

**בסיום התהליך נשלח האלט ללקוח (ששלח את ההודעה).**

## OP\_GETMAIL, OP\_DELETEMAIL

במצב זה תוכנת השרת תקרא שני בתים נוספים מהלקוח, ותפרש אותם כערך נומרי n המסמל אינדקס למייל בתיבה של הלקוח. לאחר מכן תוכנת השרת תבדוק אם האינדקס הזה חוקי (קיימים בתיבה לפחות n מיילים, וגם המייל ה-n אינו מחוק). אם כן, עבור DELETE\_MAIL יימחק המייל הרלוונטי, עבור GET\_MAIL יישלח פרינט מהשרת המכיל את המייל בפורמט:

From: sender

To: recipients

Subject: subject

Text: text

במידה והאינדקס לא חוקי, יישלח פרינט מהשרת המודיע על כך.

בכל מקרה בסוף הפעולה נשלח ללקוח האלט.

## OP\_QUIT

במצב זה תוכנת השרת סוגרת את הסוקט עם הלקוח, **מעדכנת את ה-FD שלו ל-(1-), מסירה אותו מרשימת ה-FD הפעילים** וחוזרת להאזין ללקוחות **אחרים**. ~~חדשים המנסים להתחבר.~~

## OP\_COMPOSE

במצב זה תוכנת השרת מאזינה ל-3 דאטות: targets, subject, content. השרת יוסיף מייל חדש עם הנתונים האלו למשתנה המיילים הגלובלי mails, ולאחר מכן יסרוק את מחרוזת הטרגטס שבאופן כללי מהצורה target1,target2,…,target, ולכל נמען בה, יצרף את האינדקס למייל החדש ב mails, לאינבוקס הפרטי של הנמען. בסיום יישלח ללקוח פרינט שהמייל נשלח ואז האלט. במידה וחלק מהנמענים לא קיימים במערכת הדבר יצויין בפרינט שנשלח.

# טיפוסי הנתונים המרכזיים בשרת המיילים

