

ש.ב. 1 – רשתות תקשורת מחשבים

שמות המגישים:

נוה טל 208518456 navetal@gmail.com

עמית עובדיה 208520825 amitovadia1@gmail.com

קרן סולודקין 316127893 keren.solodkin@gmail.com

תיאור הפרוטוקול שמימשנו:

הפרוטוקול נעשה מעל TCP. ראשית, מיד אחרי יצירת החיבור, נשלחת הודעת הWELCOME. הלקוח שולח את שם המשתמש ובתמורה מקבל OK, לאחר מכן שולח סיסמה ובתמורה מקבל Y (אם בסדר) או N (אם לא בסדר). אם קיבלנו N – הקישור נקטע. אם קיבלנו Y – החיבור נוצר.

יש 5 סוגי בקשות:

1. הלקוח שולח לשרת את המחרוזת "1" ובתמורה מקבל את הINBOX שלו כפי שמתואר בתרגיל.
2. הלקוח שולח לשרת את המחרוזת "2" ומספר כלשהו בן לכל היותר 5 ספרות ובתמורה מקבל ההודעה המתאימה לו.
3. הלקוח שולח לשרת את המחרוזת "3" ומספר כלשהו בן לכל היותר 5 ספרות ובתמורה ההודעה שזה מספרה תימחק והשרת ישלח ללקוח OK.
4. הלקוח שולח לשרת את המחרוזת "4" ומתנתק, לאחר מכן השרת מנתק את החיבור מהלקוח וממתין ללקוחות נוספים.
5. הלקוח שולח לשרת את המחרוזת "5" ומקבל OK, שולח מחרוזת של הנמענים ומקבל OK, שולח את הנושא ומקבל OK, שולח את תוכן ההודעה ומקבל OK.

התוכניות וכיצד להריץ אותן:

יש לקרוא לMAKEFILE על ידי הפקודה make all. לאחר מכן נוצרים שני קבצים, האחד mail_server והשני mail_client. להרצת mail_server יש צורך בפרמטר של רשימת המשתמשים, ופרמטר אופציונלי נוסף של מספר הפורט בו משתמשים, הערך הדיפולטיבי עבורו הוא 6423. mail_client מקבל שני פרמטרים אופציונליים (hostname port – אם מעוניינים לתת port אז יש צורך להעביר hostname לפני כן), הערכים הדיפולטיביים הם hostname = localhost, port = 6423.

הנחות:

אנחנו מניחים שהלקוח מכניס את השדות בסדר הנכון (ראה פרוטוקול).

אנחנו מניחים שכשהוא נדרש, הלקוח רושם את ערך השדה לפני ערכו (לדוגמה (to: amit,keren,nave).

אנחנו לא מתחייבים על פעולת התוכנית אם ההנחות לא מתקיימות.

מבנה התוכניות:

mail_client:

readInto – פעולה עוטופת לfgets, מקבלת מערך של תוים ומספר n וקוראת מstdin (עד לסוף שורה או EOF), לתוך המערך לכל היותר n תוים.

readField – פעולה שמקבלת מערך של תוים וקוראת מstdin את מספר התוים שהוא גודל מערך הקלט ומחזירה 0 אם יש התאמה ביניהם, ו-1 אחרת.

sendAll, recvAll – כפי שראינו בכיתה.

handleError – מדפיס שקרתה שגיאה.

startsWith – הפעולה מקבלת שני מערכי תווים ומחזירה 1 אם השני הוא רישא של הראשון ו0 אחרת. (חישוב הרישא לא כולל את null terminator שבסוף השני)

mail_server:

sendAll, recvAll – כפי שראינו בכיתה.

struct user – מייצג משתמש, בעל שני שדות. שם ומספר זיהוי.

struct mail – מייצג אימייל, בעל שישה שדות. מוען, נמענים, מספר נמענים, נושא, תוכן, מספר סידורי.

האופן בו השרת עובד:

ראשית, השרת בודק את מספר הפרמטרים שהועברו לו ומגדיר את portn בהתאם. לאחר מכן, השרת מאתחל את הסוקט ועושה בינד. השרת מאתחל את רשימת המשתמשים מהקובץ, מתחיל להאזין ללקוחות ונכנס ללולאה אינסופית. בתוך הלולאה, הסרבר מאתחל את המערכים בהם נשתמש ומחכה שיתחבר אליו לקוח. לאחר התחברות, הוא שולח WELCOME ומקבל שם משתמש וסיסמה. הוא בודק האם הם מתאימים למשתמש קיים ואם כן הוא מאשר את כניסתו וטוען את המידע שלו. לאחר מכן הסרבר מקבל מהלקוח פקודות ומבצע אותן עד אשר מתקבלת הפקודה QUIT.

מקרי קצה שלא היו חלק מהגדרת התרגיל:

אם לקוח מבקש למחוק אימייל שלא קיים, השרת לא עושה דבר.

במקרה בו הלקוח מבקש לקרוא אימייל שלא קיים, הוא מקבל הודעת שגיאה שמסבירה שהוא לא קיים.

אם ברשימת הנמענים של האימייל ישנם משתמשים שאינם קיימים, השרת מתעלם מהם.