

רשתות תקשורת מחשבים

תרגיל בית 1

תיאור כללי של הפרוטוקול:

הרצון שלנו בתרגיל זה לבנות שרת שיוכל לתקשר עם יותר מלקוח אחד "במקביל", לכן בשביל לעשות זאת, אנו מגדירים שתי רשימות של לקוחות (בפועל שתי רשימות של sockets): אחת ללקוחות שהם בתהליך ההתחברות (log in) והשנייה ללקוחות שכבר התחברו, בנוסף הגדרנו מילון (dictionary) שיכיל עבור כל לקוח את ההודעה הבאה שאמורים לשלוח לו (רעיונית זה דומה למה שעשינו באחד התרגולים).

בפועל עושים לולאת while ובתוכה משתמשים ב select בשביל למנוע חסימה, ואז בכל פעם שמקבלים "בקשה" (בקשה יכולה להיות שליחת הודעה לשרת, או המתנה להודעה מהשרת, או התחברות...) מלקוח אנו בודקים אם זהו לקוח חדש, כלומר אם server_socket נמצאת ב readble, אז מקבלים את הלקוח החדש, ומוסיפים אותו לרשימת הלקוחות שבתהליך ההתחברות ובמילון רשמים לו את הודעת הברכה, נשים לב שלא שולחים מיד וזה הרעיון, היא תישלח באיטרציה הבאה בלולאת while.

אם הלקוח לא חדש, אזי ממשיכים איתו בנקודה שעצרנו בה, יכול להיות שהוא בתהליך התחברות או בתהליך שליחת שאילתה לשרת...

תיאור לגבי ההדפסות וההתמודדות עם השגיאות: (זה גם מסביר מה מצופה מהלקוחות מה הוא צריך לבדוק ומה לא...)

- אם הלקוח שולח מספר ארגומנטים לא תואם (או שמקבלים כלום או שמקבלים גם hostname וגם port) אז נדפיס "Invalid input" ונתנתק ע"י return, גם בשרת אם קיבלנו מספר ארגומנטים לא תואם (קובץ תמיד נקבל אז או שנקבל גם host או בלי) אז נדפיס אותה הודעה רק הפעם בשרת ונתנתק.
 - אם יש בעיה אצל הלקוח בהתחברות לשרת, זה יכול להיגרם כתוצאה מכך שבקובץ הלקוח קיבלנו למשל host לא תקין או port לא תקין אז מדפיסים הודעה מתאימה "Error connecting to the server" ומתנתקים ע"י return, זה אמור להיות בקובץ הלקוח.
 - אם יש בעיה בפתיחת הקובץ שנשלח אז מטפלים בזה אצל השרת והוא מדפיס הודעה מתאימה "Error opening the file" ומתנתק בכלל.
 - בתהליך ההתחברות ההודעה הראשונה שהלקוח אמור לשלוח היא User: והלקוח מוודא שאכן זה קורה בקובץ הלקוח, אם לא אז הוא מדפיס "Connection close due to an incorrect log in syntax" והוא סוגר את ההתחברות מול השרת (את זה עשינו בקובץ הלקוח מכל מיני סיבות בשביל שלא יסתבך הקוד בשרת ושלא יגרום לשגיאות).
 - אם השלב הראשון שבו הלקוח שולח User: עבר בהצלחה אז אחר כך כל שגיאת סינטקס, למשל כתיבת משהו לא קשור או כתיבת Password בלי נקודתיים או רווח אחרי, או שליחת יותר מסיסמה מופרדות ברווחים (רווח אסור להיות בסיסמה), גם שליחת quit בשלב זה הינה לא חוקית (ככה התייחסנו לזה כי ככה הבנו את זה מההנחיות), תטופל בשרת ואז השרת ידאג לסגור את החיבור מול הלקוח הזה ספציפית וגם ישלח לו הודעה שהחיבור נסגר ואז הלקוח מדפיס הודעה מתאימה "connection closed due to incorrect log in syntax" וגם מתנתק (עושה return).
 - בתהליך שאחרי ההתחברות שליחת quit היא חוקית ותסגור את ההתחברות מול הלקוח ששלח אותה ספציפית, וכל שליחה בפורמט לא תקין תגרום להדפסת הודעה מתאימה: "Connection closed due to an incorrect syntax or unprovided function". טיפול בשגיאות אלה נעשה בשרת ע"י פונקציות עזר שבנינו שנסביר עליהן עוד קצת בקובץ.
 - כמובן שאם קרתה שגיאה אחרת כלשהי בשרת או בלקוח אז ה try יקלוט אותה ואז תודפס הודעה מתאימה: "An error has occurred" ויתנתק השרת/הלקוח.
- הערה:** כמעט כל שגיאות הסינטקס שהלקוח יכול לגרום להן, כלומר שישלח הודעה בפורמט לא תקין או לא מתאימה לשלב שהוא בו נמצא טופלה ע"י השרת, היה אפשר לטפל בזה ע"י הלקוח כלומר בקובץ הלקוח והאמת שבהתחלה עשינו את זה בלקוח אבל שינינו הכל שיהיה בשרת על אף שזה יותר קשה מבחינת קוד, והסיבה לכך, היא שאנו חושבים שרעיונית זה הדבר הנכון שצריך לעשות, כי

ככה אפשר בקלות לעשות קובץ לקוח שיתקשר עם השרת, ואז השרת הוא זה מי שיקבע אם יש שגיאות ובעיות בהודעות שנשלחות, זה גם לא הגיוני לבקש מכל לקוח שבא לתקשר את השרת שהוא ידאג לשלוח הכל בפורמט תקין, זוהי אחריות השרת לוודא שהכל תקין, כי נדמיין מצב שעכשיו רוצים לבנות כמות עצומה של לקוחות שרוצים לתקשר עם השרת, אז אנו נצטרך שבכל לקוח לבדוק תקינות הקלט, אלא בדרך שאנו עשינו הבדיקה והטיפול זה נעשה במקום אחד: קובץ השרת.

תיעוד הפונקציות של השרת:

- ❖ `Check_LogIn_Msg`: הפונקציה בודקת את נכונות הקלט בעת ההתחברות ומוודא שהקלט שהוקש הוא או מהצורה `User: username` או `Password: password` גם אם עדיין לא קיבלנו קלט ומחכים מהמשתמש שיזין קלט (אז מה שיהיה מאוחסן המילון עבור לקוח זו הודעת הברכה) זה כמובן חוקי. אם הקלט תקין מחזירים `True` ואחרת מחזירים `False`.
 - ❖ `Check_Func_Msg`: הפונקציה בודקת האם אחרי ההתחברות המוצלחת הלקוח נתן לנו קלט תקין שהוא אחד מ-5 המקרים האלה:
 - ✓ `quit`
 - ✓ `is_palindrome: X` – מוודאים ש-X מספר שלם.
 - ✓ `is_primary: X` – מוודאים ש-X מספר שלם.
 - ✓ `calculate: X Y Z`: מוודאים ש-X, Z מספרים שלמים וש-Y מייצג אחת הפעולות המתוארות בתרגיל.אם קיבלנו פורמט לא תקין נחזיר `False`.
 - ❖ `SendMsgsToClients`: הפונקציה שולחת ללקוח הודעה בהתאם למצב ההתחברות הנוכחי, כלומר אם הלקוח עדיין בשלב ההתחברות אז אנחנו שולחים הודעה בהתאם לתהליך שבו הוא נמצא, תמיד בהתחלה שולחים הודעת ברכה, ואז בהתאם למה שקיבלנו קודם מהלקוח נשלח תגובה לו, אם הוא שלח שם משתמש וסיסמה נכונים בפורמט תקין נגיד לו ההודעה הרצויה, ואחרת נגיד לו שהוא נכשל אם הם לא נכונים או מנתקים אותו אם הפורמט לא תקין. וכמובן שאם התחבר בהצלחה נעדכן את רשימת הלקוחות המתחברים שלא תכלול אותו ונוסיף אותו לרשימה השנייה. וכמובן אם הלקוח בשלב השני אז נשלח תגובה אם הוא שלח פקודה בפורמט תקין או שמנתקים אותו כפי שהסברנו לעיל.
 - ❖ `GetMsgsFromClients`: הפונקציה בודקת את הקלט ששלח הלקוח ומפעילה את הפונקציה המתאימה ומעדכנת גם את המילון של ההודעות שנרצה לשלוח באמצעות הפונקציה לעיל, בהתחלה אנחנו בודקים האם השם משתמש והסיסמה שהלקוח הזין הם נכונים ותואמים, אם כן אז מעדכנים את המילון בהתאם, אחרת, אם הלקוח כבר מחובר, אז בודקים גם את תקינות הקלט ומעדכנים את המילון.
 - ❖ `StartServer()`: זו הפונקציה העיקרית בקוד אשר משתמשת בכל הפונקציות שהזכרנו לעיל ומבצעת את האלגוריתם/ הרעיון שתיארנו בתחילת הקובץ הזה.
- בקובץ הלקוח יש פונקציה עיקרית אחת `StartClient` שעושה התחברות לשרת ומקבלת הודעות מהטרמינל בשביל לשלוח לו...

התמונות מהרצת wireshark בקובץ נפרד.