

עבודת הגשה – אפליקציית צ'אט מגיש – טומי לוי הגשה – 1.1.23

פרטים טכנים

מערכת הפעלה – windows 10

רקע תאורטי

TCP (Transmission Control Protocol) – פרוטוקול תקשורת המגדיר כיצד ליצור ולתחזק חיבור רשת ותעבורת תקשורת שבאמצעותה תוכניות ויישומים יכולות לבצע תעבורת נתונים. זהו פרוטוקול של שכבת תחבורה לפי מודל ה-OSI, המשמש להבטחת מסירת נתונים מקצה אחד של ערוץ תקשורת לקצה השני בצורה אמינה ויעילה.

TCP הוא connection-oriented, כלומר יש ליצור חיבור בין שני התקנים לפני שניתן להעביר נתונים. לאחר יצירת החיבור, המכשירים יכולים לשלוח נתונים זה לזה בצורה של מקטעים (segments). TCP משתמש במספר מנגנונים כדי להבטיח מסירה אמינה של נתונים, כולל בדיקת שגיאות, בקרת זרימה ושידור חוזר של מקטעים שאבדו או שניזקו.

שרת - מחשב או מכשיר המספק שירות למחשבים או מכשירים אחרים ברשת.

לקוח - מחשב או מכשיר שמתחבר לשרת על מנת לבקש משאבים או שירותים. לקוחות משמשים בדרך כלל כדי לגשת למשאבים או לשירותים המסופקים על ידי שרתים ברשת, כגון קובץ בשרת קבצים או דף אינטרנט בשרת אינטרנט.

לקוחות ושרתים עובדים יחד כדי לספק גישה למשאבים ולשירותים דרך רשת. בעוד שהשרתים אחראים לספק את המשאבים או השירותים, הלקוחות אחראים לבקשם ולהשתמש בהם.

Threads - זרימה של קוד ביצוע (execution) בתוך תוכנית. Thread הוא נתיב ביצוע נפרד שיכול לפעול במקביל עם Threads אחרים באותה תוכנית. ה-Threads משמשים לשיפור הביצועים של תוכניות בכך שהם מאפשרים להן להריץ משימות במקביל.

ניתן להשתמש ב-Threads לביצוע משימות שונות במקביל בתוך תוכנית. למשל אצלנו, תוכנית ששולחת ומקבלת נתונים ברשת תשתמש ב-Thread נפרד כדי לטפל בכל חיבור נכנס. זה מאפשר לתוכנית לטפל במספר חיבורים בו זמנית, ולשפר את הביצועים הכוללים שלה.

Mutex - (בעברית זה קיצור של "אי הכללה הדדית") הוא אובייקט סנכרון השולט בגישה למשאב משותף בתוכנית מרובה הליכים. הוא משמש כדי למנוע מ-Threads מרובים לגשת למשאב המשותף בו זמנית, מה שעלול להוביל לתנאי מרוץ ולבעיות אחרות.

הרעיון הבסיסי מאחורי mutex הוא שהוא מאפשר רק ל-Thread אחד לרכוש מנעול על mutex בכל זמן נתון. כאשר Thread רוכש את המנעול, הוא יכול לגשת למשאב

המשותף. כאשר הוא מסתיים, הוא משחרר את הנעילה, ומאפשר ל-Threads אחרים לרכוש את המנעול ולגשת למשאב המשותף. במהלך מטלה זו היינו צריכים להבטיח את נכונות בסיס הנתונים שמאכסן את המחוברים לצ'אט מסוים.

Maps (c++) - ב-C++, מפה היא מיכל המאחסן אלמנטים בסדר ממזין, כאשר כל אלמנט מורכב מזוג מפתח-ערך. מפות מיושמות כעצים אדומים-שחורים, שהם עצי חיפוש בינאריים מאוזנים השומרים על סדר בין האלמנטים בעץ. מפות שימושיות לאחסון וגישה מהירה לנתונים המוספים לאינדקס על ידי מפתחות. במהלך מטלה זו ישנו אובייקט גלובלי של מפה אשר אחראי לאכסון של חדרי הצ'אט הקיימים כאשר המפתח של העץ הוא סטרינג רנדומלי המורכב מ 5 אותיות באנגלית (גדולות/קטנות או גם וגם) המשמש כ ID של החדר.

Socket - נקודת קצה תקשורת לשליחה או קבלה של נתונים ברשת מחשבים. זהו מנגנון המאפשר לתוכנית הפועלת במכשיר אחד לשלוח או לקבל נתונים לתוכנית הפועלת במכשיר אחר או ממנה.

קוד שרת

בצד השרת אנחנו נמצא 2 קבצים: קוד מקור וקובץ קוד בשם `useful_funcs.h` המכיל פונקציות שיצרתי עבור השרת למשימה זו. התבקשנו שהאפליקציה תדע לעשות שלושה דברים:

1. חיבור לחדר צ'אט קיים – הלקוח באופציה זו מתבקש להזין ID תקין של צ'אט קיים וסיסמה של החדר המבוקש. במידה והזיהוי הושלם בהצלחה, השרת יבקש גם את השם של הלקוח ויחבר אותו לחדר הצ'אט. בגרסה של תוכנה זו שיצרתי ללקוח יש 3 ניסיונות להקלדה תקינה של ID קיים ושלושה ניסיונות להקלדה תקינה של סיסמת החדר הקיים, במידה ולא יעמוד בניסיונות אלו השרת מנתק באופן אוטומטי את הלקוח בשל זיהוי לקוי בתהליך התחברות. (קורה בקבלת פקודה מלקוח של `$create`)

2. יצירת חדר צ'אט – הלקוח באופציה זו מתבקש להזין סיסמה לחדר הצ'אט החדש שהוא רוצה ליצור, לאחר הזנת הסיסמה הלקוח מקבל למסך שלו את ID היחודי שיוצר עבור חדר זה ואז השרת יבקש את שמו של הלקוח ויחבר אותו לחדר שיצר. בפונקציה האחראית על יצירת חדר הצ'אט החלטתי כי מצב בו חדר יהיה ללא סיסמה. לכן, לקוח אשר ינסה "לבדוק" את המערכת יקבל "הודעה מעניינת" ויקבע לו סיסמה דיפולטית לחדר שנקראת `TESTER`. מקרה זה גם קורה כאשר הלקוח מבצע טרמינציה של התוכנה מצידו במהלך התהליך. (קורה בקבלת פקודה מלקוח של `$login`)

3. יציאה מממשק הלקוח – בחירה באופציה זו תבצע טרמינציה של הלקוח מהשרת, זריקת `socket` עליו היה מחובר וסגירת הממשק בצד הלקוח. תוספת שלי למשימה זו

הייתה שגם כאשר הלקוח מחובר לצ'אט ויחליט שהוא רוצה לסגור את התוכנה, הלקוח יכול להקליד את אותה פקודה והשרת יבדוק אם הלקוח הקליד בדיוק את אותה הפקודה (כלומר שהבאפר מכיל אך ורק את הפקודה הדרושה) השרת יבצע את היציאה מממשק הלקוח. (הפקודה לא מופיע כפונקציה בפני עצמה אך ניתן לראות אותה בקוד המקור תחת מצבים בהם הוקלד \$exit\$)

תוספות של פונקציות בקובץ זה:

1. generate_random_string() – במהלך יצירת ID ייחודי עבור כל חדר שניצור ראיתי כי בדוגמא השתמשו ברצף של 4 תווים של מספרים. לאחר מחשבה בנושא הגעתי למסקנה שזה יכול להיות טיפה מגביל עבור כמות החדרים הייחודים שיכולים להיות (10000 קומבינציות שונות). רציתי לראות איך אני יכול להגדיל את כמות החדרים שיש לי ובו זמנית גם לדאוג טיפה ל"פרטיות המשתמש" (אני יוצא מנקודת הנחה שהמשתמש לא רוצה גם אנשים לא רצויים בחדר הצ'אט שלו ולכן רציתי ליצור "הגנה נוספת" כדי להקשות על אפשרות של משתמש לנחש את הID של חדר צ'אט מסוים). פונקציה זו נועדה לפתור בעיה זו.

הפונקציה בעצם תגדיל לנו ID בצורת סטרינג המורכב מ5 תווים בשפה האנגלית (גם תווים רגילים וגם CAPS LOCK) וזה יהיה הID של החדר שנוצר. בחישוב קצר ניתן לראות כי הגדלנו משמעותית את כמות החדרים האפשריים עם ID ייחודי (26 אותיות באנגלית + 26 אותיות באנגלית עם CAPS LOCK לכן 52 אותיות אפשריות. לכן כמות הקומבינציות האפשריות היא 52 בחזקת 5: 140,608 קומבינציות שונות.) כמובן שבקלות ניתן להבין שיהיה הרבה יותר קשה גם "לנחש" את הקומבינציה הנכונה.

2. מבנה הנתונים ChatRoom – מבנה נתונים ייחודי אשר יצירתי עבור שרת זה. מבנה הנתונים בעצם משמש כאיגוד לכל השדות הנדרשים לקיום של "חדר צ'אט" ונעבור עליהם רגע בצורה סיסטמטית:

- שדה הלקוחות – ווקטור של Sockets של לקוחות אשר מחוברים לאותו חדר צ'אט.
- שדה פוינטר mutex – כיוון שמבנה הנתונים יהיה תחת שינויים תמידיים בשדה הלקוחות (לקוח מתחבר/מתנתק) נרצה להבטיח שנשמרת נכונות המידע עבור כל הלקוחות המחוברים לאותו חדר צ'אט. לכן כאשר Thread מסוים של לקוח מנסה לשנות משהו במבנה הנתונים של החדר המדובר, נדאג כך להבטיח את תקינות המידע עבור שאר הלקוחות בחדר.
- שדה הסיסמה – טיפוס נתונים מסוג סטרינג בכדי לאכסן את הסיסמה של החדר.
- שדה הID – טיפוס נתונים מסוג סטרינג בכדי לאכסן את הID של החדר.

(הסבר אקסטרה – למה דווקא פוינטר של mutex ולא פשוט האובייקט עצמו? לא יהיה יותר פשוט? . התשובה מפתיעה אבל דווקא לא! בשביל להבין גם למה לא צריך להבין כי mutex חייב להיות אובייקט שלא ניתן להעברה או העתקה. כאשר אנחנו מנסים לקרוא לפונקציה של insert או emplace עבור האובייקט מסוג map שמאכסן לנו את כל החדרים, הקומפילר מנסה לקרוא ל pair או make_pair כדי להכניס את הערכים לmap אולם השדה של mutex בתוך המבנה הנתונים שהגדרנו לא יהיה ניתן להעברה או העתקה ולכן ניסיון לקמפל את הקוד במצב זה יזרוק לנו שגיאה 2660C . הדרך בעצם להתגבר על בעיה זו להגדיר את השדה כפוינטר של mutex ואז לבנות שני קונסטרוקטורים במבנה הנתונים. האחד יחליף את הקונסטרוקטור הדיפולטיבי ובו נבצע הקצאה דינאמית של mutex והשני יהיה קונסטרוקטור עם פרמטרים, אשר משמש אותנו בקוד זה בכדי להחל את הסיסמה של החדר ולשמור את ה ID של החדר במבנה הנתונים. הסיבה לכך שזה באמת פותר את הבעיה שלנו היא שפוינטר מבחינת ++C הוא בעצם שקול ל int* שהרי זה פשוט כתובת בזיכרון ולכן pair ו make_pair לא זורקים לנו שגיאה 2660C. יותר מזה אני אדגיש שאין לי צורך באמת בקוד זה לכתוב דיסטרוקטור שיטפל בהקצאה הדינאמית כי כל עוד השרת עובד אני לא באמת רוצה למחוק חדרים, פשוט לא ביקשתם ממש פונקציה זו אבל בעתיד אם אחליט להרים את הקוד הזה יותר לעומק אני כמובן אכין פונקציה כזו ואטפל בדיסטרוקטור.)

קוד המקור

ניתן לפרק את קוד המקור לכמה שלבים

* אתחול winsock – בכדי לאתחל לשימוש בגרסה 2.2

* יצירת socket לשרת – בשלב זה אנחנו מייצרים לשרת socket שיהיה "מאזין" עבור חיבורים נכנסים של לקוחות, פה אפשר לראות כי קבענו את הsocket לעבוד בצורה של IPPROTO_TCP אשר זה פרוטוקול TCP סטנדרטי. בנוסף קיבענו את הsocket לעבוד עם AF_INET שזה בעצם למשפחת IP מצורה של IPv4. (בנוגע ל SOCK_STREAM זה פשוט בכדי לעבוד בפרוטוקול TCP עבור ספרייה זו)

* bind – לאחר שהsocket המאזין נוצר בהצלחה נרצה ל"לקשר" אותו עבור הכתובת הלוקאלית של המחשב ברשת ו"לקשר" אותו עבור פורט מספר 5000.

* הפעלת המאזין לביצוע האזנה – פשוטו כמשמעותו. בנוסף נכניס למאזין את גודל התור שלו.

* השרת עצמו והthreads – כאן צריך לשים לב שהקוד מתפצל לכמה threads שונים. ה thread הראשון נקרא serverThread. האחריות שנתנה לו היא כטיפול בשרת בלקוחות שמתחברים לשרת ולעגן חיבורים תקינים שלהם לטיפוס מסוג Socket. משם אנו פותחים Thread נוסף בשם clientThread. אותו Thread מתעסק בפונקציונליות של השרת כלפי

עבור כל לקוח. פה מתבצע הצגת המסך הראשי (בנוסף פה גם בפונקציות החיבור והיצרת חדר מוחזר לנו הערך של הID של החדר אליו הלקוח התחבר ונשמר במשתנה שישמש לשליחת ההודעות) ושליחת ההודעות לצ'אט אליו הלקוח מחובר. ובמידה והלקוח ביקש התנתקות או התנתק "בכוח" (סגירת טרמינל לקוח) יתבצע ניקוי ויציאה – זריקת Socket בשל התנתקות הלקוח.
*ניקוי ויציאה – זריקת Socket המאזין וסגירת התוכנה.

כמובן שלכל שלב בקוד המקור ניתן למצוא טיפול במקרה קצה.

קוד לקוח

צד הלקוח מכיל רק קובץ מקור. כן ההסבר יותר פשוט לכן אסביר גם פה בצורה של שלבים סיסטמית:

- * אתחול winsock – בכדי לאתחל לשימוש בגרסה 2.2
- * יצירת socket ללקוח – בשלב זה אנחנו מייצרים ללקוח socket פה אפשר לראות כי קבענו את הsocket לעבוד בצורה של IPPROTO_TCP אשר זה פרוטוקול TCP סטנדרטי. בנוסף קיבענו את הsocket לעבוד עם AF_INET שזה בעצם למשפחת IP מצורה של IPv4.
- * bind – לאחר שהsocket נוצר בהצלחה נרצה ל"לקשר" אותו להגדרות הדרושות לחיבור לצד השרת.
- * ניסיון חיבור הלקוח לשרת – פשוטו כמשמעו, הלקוח שולח בקשת התחברות לשרת, במידה ולא יצליח ליצור התקשרות עם השרת, יזרוק שגיאה וידפיס הודעת שגיאה ללקוח.
- * פיצול לThread נפרד – כאן ניצור Thread שיהיה קשור לקבלת מידע מהשרת. (שמו בקוד: receiveThread) בעצם הוא יאזין למידע שנשלח מהשרת, ויפעל בהתאם אם השרת הפסיק להגיב ללקוח ויש תקלה\התנתקות מהשרת או שזה בעצם הודעת צ'אט שצריך להדפיס למסך.
- * שליחת מידע לשרת – כאן אנו ממתינים ללקוח שיכתוב הודעות לטרמינל. כמובן שלפני שהמערכת תחשוב שהלקוח פשוט שולח הודעה רגילה, היא תבדוק האם הלקוח הקליד את אחת הפקודות המוסכמות (\$login\$ \$create\$ \$exit\$) ותטפל בהתאם. (הטיפול יהיה דיי כמו לשלוח הודעה רגילה חוץ מב\$exit\$ ששם הוא זורק את הלקוח מהלולאה וגורם לטרמינציה של הטרמינל של הלקוח)
- * ניקוי ויציאה – זריקת Socket לקוח וסגירת התוכנה.

אילוסטרציה:

משתמש יוצר חדר:

```
C:\Users\user\source\repos\chat_server\Debug\chat_server.exe
Listening for incoming connections on port 5000...
Accepted connection from 127.0.0.1:54700
room id:FmUYD has password:a

C:\Users\user\source\repos\chat_client\Debug\chat_client.exe
Connected to the server
Wellcome to Tommy's Chat App!
what would you like to do today?
$login$ to login to a chatroom (fail to login and you will be DC from the server.)
$create$ to create a chatroom
$exit$ to exit the app (don't worry, you can type this commend after too, it's still works)

$create$
What will be the password? :
a
Chat room id is: FmUYD
What is your name? :
tommy
```

לקוח חדש מתחבר לחדר קיים:

```
C:\Users\user\source\repos\chat_server\Debug\chat_server.exe
Listening for incoming connections on port 5000...
Accepted connection from 127.0.0.1:54700
room id:FmUYD has password:a
Accepted connection from 127.0.0.1:54702

C:\Users\user\source\repos\chat_client\Debug\chat_client.exe
Connected to the server
Wellcome to Tommy's Chat App!
what would you like to do today?
$login$ to login to a chatroom (fail to login and you will be DC from the server.)
$create$ to create a chatroom
$exit$ to exit the app (don't worry, you can type this commend after too, it's still works)

$create$
What will be the password? :
a
Chat room id is: FmUYD
What is your name? :
tommy
USER2>hi
sup man?

C:\Users\user\source\repos\chat_client\Debug\chat_client.exe
Connected to the server
Wellcome to Tommy's Chat App!
what would you like to do today?
$login$ to login to a chatroom (fail to login and you will be DC from the server.)
$create$ to create a chatroom
$exit$ to exit the app (don't worry, you can type this commend after too, it's still works)

$login$
Chat room's id? (5 char.) :
FmUYD
Chat room's password?:
a
welcome to chat room with id: FmUYD
What is your name? :
USER2
hi
tommy>sup man?
```

יציאה משרת:

```
C:\Users\user\source\repos\chat_server\Debug\chat_server.exe
Listening for incoming connections on port 5000...
Accepted connection from 127.0.0.1:54700
room id:FmUYD has password:a
Accepted connection from 127.0.0.1:54702
Disconnect connection 127.0.0.1:54702
Client disconnected

C:\Users\user\source\repos\chat_client\Debug\chat_client.exe
Connected to the server
Wellcome to Tommy's Chat App!
what would you like to do today?
$login$ to login to a chatroom (fail to login and you will be DC from the server.)
$create$ to create a chatroom
$exit$ to exit the app (don't worry, you can type this commend after too, it's still works)

$create$
What will be the password? :
a
Chat room id is: FmUYD
What is your name? :
tommy
USER2>hi
sup man?
_
```