רשתות תקשורת

אופן הרצת התוכנית:

- אתחול השרת מתבצע באמצעות הפקודה:
./seker_server users_file dir_path [port]
כאשר:
  - users_file - נתיב לקובץ טקסט שמכיל שמות משתמשים וסיסמה לכל משתמש.
  - dir_path – נתיב לתיקייה בה השרת ישתמש לאחסון מידע.
  - port – פרמטר אופציונלי- ערך ברירת מחדל 1337.
- אתחול הלקוח מתבצע באמצעות הפקודה:
./seker_client [hostname [port]]
כאשר:
  - hostname – פרמטר אופציונלי - שם או כתובת IP של השרת – ערך ברירת מחדל: localhost
  - port – פרמטר אופציונלי- ערך ברירת מחדל 1337.

תיאור הפרוטוקול:

כל הודעה שנשלחת בין השרת והלקוח מורכבת מהחלקים הבאים:

Header

type 1 byte	is_last 1 byte	data_len 4 bytes	data

כאשר:

- type – מזהה עבור סוג ההודעה – 1 byte
- is_last – האם זו ההודעה האחרונה ברצף- 1 byte
- data_len - אורך ההודעה – 4 bytes
- data- תוכן ההודעה- אורך מקסימלי של MAX_DATA_LEN
  - אם גודל ההודעה גדול מ MAX_DATA_LEN , ההודעה תשלח ותתקבל בחלקים.
  - כאשר הלקוח שולח לשרת בקשה המכילה פרמטרים, הפרמטרים יופרדו ע"י '\n'.

אופן התקשורת בין השרת והלקוח:

- כשלקוח חדש מתחבר לשרת, השרת שולח לו ברכת שלום.
ההודעה מיוצגת בצורה הבאה:

WELCOME type	1 is_last	data_len data_len	WELCOME_MSG data

- הלקוח שולח לשרת הודעה מסוג LOGIN המכילה שם משתמש וסיסמה בפורמט הבא:

LOGIN type	1 is_last	data_len data_len	"user_name\npassword" data

השרת מפרסר את פרטי ההתחברות ומוודא את נכונותם.
אם פרטי ההתחברות נכונים נשלחת ללקוח הודעה מסוג LOGIN_SUCCESS בפורמט הבא:

LOGIN_SUCCESS type	1 is_last	data_len data_len	"Hi %s, good to see you.\n" data

אחרת, נשלחת הודעה מסוג LOGIN_FAIL בפורמט הבא:

LOGIN_FAIL type	1 is_last	data_len data_len	"Failed to login.\n" data

- לאחר שההתחברות הצליחה, הלקוח יכול לשלוח לשרת בקשות ולקבל ממנו תשובות.

<u>הפקודות האפשריות:</u>

- **list_of_courses**:

פקודה זו מציגה רשימה של כלל הקורסים הקיימים.

הלקוח שולח לשרת את ההודעה הבאה:

LIST_OF_COURSES type	1 is_last	0 data_len	NULL data

השרת מקבל את ההודעה, ושולח ללקוח רצף הודעות מסוג LIST_OF_COURSES,

כאשר תוכן ההודעות הוא רשימת הקורסים הקיימים.

השדה is_last של ההודעה האחרונה ברצף נקבע להיות 1.

- "**add_course** course_number "course_name":

פקודה זו מאפשרת למשתמש להוסיף קורס למאגר הקורסים בשרת.

הלקוח שולח לשרת את ההודעה הבאה:

ADD_COURSE type	1 is_last	data_len data_len	course_number\ncourse_name data

השרת מקבל את ההודעות ואם הקורס אינו קיים מוסיף אותו למערכת.

בנוסף, שולח ללקוח הודעה מסוג ADD_COURSE ,

שתוכנה הוא "course_number exists in the database!" אם הקורס כבר קיים במערכת,

אחרת, תוכן ההודעה הוא "course_number added successfully.".

בנוסף, אם הקורס נוסף בהצלחה, נשלחת הודעה שתוכנה הוא "A new course was just added!" לכל המשתמשים האחרים שמחוברים כרגע.

- "**rate_course** course_number rating_value "rating_text":

פקודה זו מאפשרת ללקוח לדרג קורס קיים.

הלקוח שולח לשרת את ההודעה הבאה:

RATE_COURSE type	1 is_last	data_len data_len	course_number\nrating_value\nrating_text data

השרת מקבל את ההודעות,

אם הקורס שצוין קיים - מוסיף את הדירוג לקובץ המכיל את הדירוגים של הקורס הנ"ל,

אחרת – נשלחת הודעה שתוכנה הוא: "The course doesn't exist in the database!\n".

- **get_rate** course_number:

פקודה זו מציגה את כלל הדירוגים הקיימים עבור הקורס.

הלקוח שולח לשרת את הבקשה הבאה:

GET_RATE type	1 is_last	data_len data_len	course_number data

השרת מקבל את ההודעה, וכתגובה שולח ללקוח הודעות מסוג GET_RATE,

אם הקורס שצוין קיים - תוכן ההודעות הוא כלל הדירוגים הקיימים לקורס,

אחרת – נשלחת הודעה אחת שתוכנה הוא : "The course doesn't exist in the database!\n".

- "**broadcast** "message":

פקודה זו מאפשרת שליחת הודעה לכלל המשתמשים המחוברים כרגע למערכת.

הלקוח שולח לשרת את הבקשה הבאה:

BROADCAST type	1 is_last	data_len data_len	message data

השרת שולח לכלל הלקוחות שמחוברים כרגע את ההודעה:

"user_name sent a new message: …"

כאשר במקום user_name הוא שם המשתמש של המשתמש ששלח את ההודעה.

- **quit:**
פקודה זו מנתקת את הלקוח מצד השרת.
הלקוח שולח לשרת את ההודעה הבאה:

QUIT type	1 is_last	0 data_len	NULL data

בתגובה, השרת מסיים את החיבור עם הלקוח.

- במידה והמשתמש מכניס פקודה לא חוקית, מודפס "Illegal command.\n".

<u>התמודדות עם שגיאות:</u>

במקרה של שגיאה בקריאת מערכת באחד הצדדים, מוצגת הודעת שגיאה מתאימה וריצת התוכנית מסתיימת.

מבנה התוכנית:

- **seker_client.c**:
מכיל את הקוד עבור צד הלקוח.
יוצר חיבור מול השרת לפי הפרמטרים שצוינו או פרמטרי ברירת המחדל.
מאפשר קבלת מידע מהשרת ב push בעזרת שימוש ב select עם ה file descriptors של STDIN ושל ה socket מול השרת.
כאשר STDIN מוכן לקריאה, התוכנית מקבלת את הקלט מהמשתמש, מפרסרת אותו לפקודה המתאימה, ושולח את הבקשה לשרת.
כאשר ב socket מול השרת מוכן לקריאה, מציגה את תגובת השרת לבקשה או הודעה שנשלחה ב push.
- **seker_server.c:**
מכיל את הקוד עבור צד השרת.
מאזין לחיבורים נכנסים, ומשרת את הלקוחות במקביל בעזרת שימוש ב select.
מקבל בקשות מהלקוח, מבצע אותן לפי הצורך, ושולח אליו תשובות.
- **Command.h, Command.c:**
  - מגדיר struct בשם Command המכיל את סוג הפקודה, האם הארגומנטים שסופקו לפקודה תקינים, ומחרוזת המכילה את הפרמטרים שהועברו לפקודה.
  - מגדיר enum בשם Command_Type המכיל את סוגי הפקודות האפשריות.
  - מפרסר מחרוזת (שהתקבלה כקלט מהמשתמש) אל אובייקט מסוג Command.
  אם הפקודה אינה חוקית, השדה של סוג הפקודה מוגדר להיות CMD_INVALID.
  אחרת, הוא מוגדר בהתאם לסוג הפקודה.
  במידה והפקודה מצפה לפרמטרים, נבדר האם הפרמטרים חוקיים, ואם כן, השדה validArgs מוגדר להיות true, והשדה args מכיל את הפרמטרים משורשרים ע"י '\n'.
- **Message.h, Message.c:**
  - מגדיר struct בשם Header המכיל את שדות ה metadata של ההודעה: סוג ההודעה, האם זו ההודעה האחרונה ברצף, ואורך ההודעה.
  - מגדיר struct בשם Message המייצג את ההודעה הנשלחת בין הצדדים: מורכב מ Header ומערך של charים באורך MAX_DATA_LEN עבור תוכן ההודעה.
  - מגדיר enum בשם MESSAGE_TYPE עבור סוגי ההודעות האפשריות.
  - מכיל את הפונק' עבור שליחת וקבלת ההודעות.

הגדרת קבועים:

- CONNECTION_QUEUE_SIZE= 100: מס' מקסימלי של לקוחות היכולים להמתין לחיבור עם השרת.
- MAX_DATA_LEN = 2048: אורך מקסימלי של הודעה.
- MAX_LINE_LEN = 4096: אורך מקסימלי של שורת קלט מהמשתמש.
- MAX_PATH= 1000: אורך מקסימלי של הנתיב לתיקייה המשמשת לאחסון.
- MAX_USERS = 25: מס' מקסימלי של לקוחות במערכת.
- MAX_RATING_LEN = 1024: אורך מקסימלי של דירוג טקסטואלי.
- MAX_RATING_VAL_LEN = 3: מס' הספרות המקסימלי של דירוג מספרי (בין 0 ל-100).
- MAX_COURSE_NAME_LEN = 100: אורך מקסימלי של שם קורס.
- MAX_COURSE_NUMBER_LEN = 4: מס' הספרות המקסימלי של מס' קורס (בין 0 ל-9999).
- MAX_USER_LEN = 15: אורך מקסימלי של שם משתמש.
- MAX_PASS_LEN = 15: אורך מקסימלי של סיסמה.