

## תרגיל סוקטים מתקדם - כתיבת צ'אט מרובה משתמשים

ברק גונן

סרטון הסבר לתרגיל:

<https://youtu.be/bWU6TfHbf1g>

הערה חשובה- בתרגיל זה תידרשו לבצע פקודות נוספות על מה שמודגם בסרטון. הסרטון מדגים את הפונקציונליות הבסיסית.

כיתבו שרת ולקוח שיאפשרו צאט בין משתמשים. הגישו את קובץ השרת, קובץ הלקוח, קובץ הפרוטוקול.

בידקו שהשרת עובד עם מספר לקוחות שמתקשרים ביניהם. עיברו על דוגמת ההרצה וודאו שהריצה שלכם מתנהלת באופן זהה.

הפקודות בהן תומך הלקוח:

```
# NAME <name> will set name. Server will reply error if
duplicate
# GET_NAMES will get all names
# MSG <NAME> <message> will send message to client name
or to broadcast
# BLOCK <name> will block a user from sending messages
to the client who sent the block command
# EXIT will close client
```

הסבר:

**פקודת NAME** מאפשרת לקבוע שם ללקוח שלנו. לדוגמה NAME A יקבע את השם A עבור הלקוח, כך שכעת הודעות שמיועדות ל-A יועברו אליו על ידי השרת. השרת יענה עליה ב-HELLO עם השם שנקבע. לדוגמה HELLO A. אם השם A תפוס כבר על ידי לקוח אחר, השרת יענה שהשם תפוס.

**פקודת GET\_NAMES** תגרום לשרת לשלוח ללקוח את רשימת כל שמות הלקוחות שמחוברים לשרת (לא לשכוח להסיר לקוחות שהתנתקו)

**פקודת MSG** תגרום לשליחת הודעה ללקוח לפי השם שלו. לדוגמה MSG A HELLO תגרום לשליחת HELLO ללקוח A. השרת יוסיף לפני כן מי הגורם ששלח את ההודעה, לדוגמה אם B שלח את ההודעה ל-A אז אצל A יופיע B sent HELLO.

**שליחת הודעה לתפוצת כולם:** פקודת MSG עם הפרמטר BROADCAST תגרום לכך שההודעה תגיע לכלל המשתתפים בצ'אט, למעט השולח. לדוגמה אם המשתמש A כתב Hi MSG BROADCAST, אז משתמשים B, C יקבלו את ההודעה. שימו לב שמעתה השם BROADCAST אינו שם חוקי למשתמש חדש



**פקודת BLOCK** תגרום לכך שלקוח יוכל לחסום לקוח אחר משליחה אליו. לדוגמה אם B כתב BLOCK A אז ההודעות של A לא יגיעו ל-B, גם אם הן נשלחות ל-BROADCAST. אין דרישה לממש פקודת UNBLOCK.

פקודת EXIT תגרום לניתוק הלקוח וסגירת הסוקט שלו.

#### דוגמת הרצה:

במסכים הבאים ניתן לראות את הפעולות הבאות

1. לקוח התחבר
2. הלקוח קבע לעצמו את השם A
3. לקוח התחבר
4. הלקוח קבע לעצמו את השם B
5. לקוח A בדק אילו לקוחות קיימים וקיבל A B
6. לקוח B התנתק
7. לקוח A בדק אילו לקוחות קיימים וקיבל A
8. לקוח התחבר
9. הלקוח קבע לעצמו את השם B
10. לקוח A בדק אילו לקוחות קיימים וקיבל B A
11. לקוח A שלח הודעה HELLO ל-B, לקוח B קיבל את ההודעה בצירוף מסר מהשרת שההודעה נשלחה מ-A
12. לקוח B שלח הודעה HELLO ל-A
13. לקוח B התנתק
14. לקוח A התנתק

```
C:\Windows\system32\cmd.exe
C:\Users\BARAK>python c:\networks\work\networksbook\multiclient\chat_client.py
Pls enter commands

NAME A
Server sent: Hello A
GET_NAMES
Server sent: A B
GET_NAMES
Server sent: A
GET_NAMES
Server sent: A B
MSG B HELLO
Server sent: B sent HELLO
EXIT
C:\Users\BARAK>
```

```
C:\Windows\system32\cmd.exe
C:\Users\BARAK>python c:\networks\work\networksbook\multiclient\chat_client.py
Pls enter commands

NAME B
Server sent: Hello B
EXIT
C:\Users\BARAK>python c:\networks\work\networksbook\multiclient\chat_client.py
Pls enter commands

NAME B
Server sent: Hello B
Server sent: A sent HELLO
MSG A HELLO
EXIT
C:\Users\BARAK>
```

### הנחיות כלליות

בדומה לתרגיל שרת פקודות בסיסי, גם כאן עליכם להשתמש בקובץ פרוטוקול, שיכלול שדה אורך. הן השרת והן הלקוח יעשו שימוש בשדה האורך בקבלה (ולא סתם `recv(1024)`).

אם השרת מקבל הודעה שאינה קיימת, הוא יחזיר הודעה מתאימה ללקוח ששלח את ההודעה.

אם השרת מתבקש להעביר הודעה ללקוח שאינו קיים, הוא יחזיר הודעה מתאימה ללקוח ששלח את ההודעה.

### הדרכה לכתיבת השרת:

העזרו בקוד השרת שבספר הלימוד, עיקר ההבדל הוא שעליכם לכתוב פונקציה שמטפלת בהודעות שמגיעות מהלקוח. מומלץ להחזיק מבנה נתונים שיקשר בין סוקט לבין שם לקוח, מילון יכול להיות שימושי לטובת זה.

### הדרכה לכתיבת הלקוח:

כזכור הפונקציות `recv` ו-`input` הינן `blocking`, כלומר הלקוח ימתין לקבלת מידע לפני שממשיך הלאה. דבר זה ימנע ריצה תקינה של התוכנית כיוון שהלקוח עלול לא להיות מסוגל לקבל קלט משתמש עד שלא יקבל הודעה, ולהיפך- אי אפשר יהיה לקלוט הודעות עד שהמשתמש לא יסיים להזין קלט.

הפתרון מורכב משני חלקים.

במקום הפונקציה `recv` הלקוח ישתמש בפונקציה `select`. אפשר לשלוח לה `timeout` בשניות.

במקום הפונקציה `input` נייבא את המודול `msvcrt` (חלק חשוב במיומנות שלנו הוא לימוד עצמי של נושאים, ניתן למצוא הסברים ודוגמאות באינטרנט, לכן אינני מצרף הסבר) צרו קוד שקולט מהמשתמש תו-תו בעזרת הפונקציות `kbhit` ו-`getch`, מדפיס למסך ולאחר קבלת תו של ירידת שורה (`enter`) שולח את ההודעה לשרת. טיפ: בהדפסה למסך השתמשו בפרמטר `flush=True` כדי שכל תו יודפס מיידית למסך.

בהצלחה!