מה? לא הבנתי כלום?!

איך אני מתחיל? מה אני אמור לעשות? סבבה ליצור שרת... אבל... רגע, מה?

טוב, אז בשביל להבין מה אנחנו אמורים לעשות בתרגיל הזה, צריך להבין כמה דברים.

א' אנחנו רוצים לבנות שרת מולטי-ת'רד. למה? בשביל שיהיה שני לקוחות שמתקשרים ביניהם, אנחנו צריכים את היכולת קודם כל שיהיה שני לקוחות מחוברים בו זמנית. ככה אנחנו נוכל בתור השרת להעביר בין שני הלקוחות שלנו את ההודעות.

אז שלב ראשון, להפעיל שרת ולהפוך אותו למולטי-ת'רד.

איך, אל תסתבכו יותר מידי, יש בנספחים של השיעור (שיעור 13) את הדוגמת קוד של השרת שהוצג בשיעור. קחו אותו ותריצו אותו על המחשב שלכם, אם הוא עובד אפשר להתחיל לנסות להפוך אותו למולטי-ת'רד... (אם הוא לא, זה הזמן לשאול את גוגל "מה הוא רוצה ממני"... כמובן שגם יהל וגם אני נחשבים גוגל במקרא הזה...)

אז יש לנו שרת מולטי-ת'רד, מה עכשיו?

מושלם! אבל עכשיו בשביל להבין איך מתקדמים מפה, צריך להבין איך השרת של מגשימים עובד. כמו בכל שרת, בשביל שהלקוח והשרת יכולו להבין אחד את השני, עושים פרוטוקול מוגדר מראש, ששני הצדדים מדברים לפי החוקים שלו. ככה גם השרת וגם הלקוח יוכלו להבין אחד את השני. אז בשביל להבין את פרוטוקול של מגשימים, בא ננסה להבין מה אנחנו צריכים. תחשבו על זה שאתם הלקוח, ברגע שאתם חיברתם סוקט לשרת, מה הדבר הראשון שאתם רוצים לומר לו? מצוין, מי אנחנו. אז בשביל זה יש את **הבקשת התחברות** אנחנו זוכרים אותה, וממשיכים אלאה. לאיזה תשובה הייתם מצפים מהשרת? מה הלקוח רוצה לדעת עכשיו? ברור! הוא הרגע התחבר, הוא רוצה לדעת מי כבר מחובר!

אז אמרנו ככה: הלקוח שולח את השם שלו, והשרת מחזיר לו את כל מי שמחובר. יפה.

באו נראה איך זה נראה באמת. ונגיד שללקוח שלנו קוראים בוב. ההודעה תראה ככה:

"200 03 bob"

למה לא לשלוח פשוט "bob"? כי אנחנו קודם כל צריכים לסמן לשרת שזאת הודעת התחברות. ואחרי זה אנחנו פשוט שולחים את השם. אז למה אנחנו צריכים את ה"03" שלפני? זה כי אנחנו לא יכולים לדעת כמה לקראו מהשרת (זה יכול להיות אין סוף, אנחנו חייבים לדעת מתי להפסיק לקרוא) ולכן אנחנו נכתוב לפני בשני תווים מה האורך של המחרוזת הבאה.

שאלה להבנה: מה אורך השם הכי ארוך שאנחנו יכולים לשלוח לשרת? יפה 99 למה? כי זה מספר הבתים שיש לנו. זה מספר קבוע שהוא 2, ואנחנו גם לא צריכים יותר מזה...

אוקי, אז מה אמרנו שהשרת צריך להחזיר? אוקי נכון, את כל המשתמשים. אז ככה זה נראה:

"101 00000 00 00009 bob&alice"

למה יש שם מלא אפסים? את זה תבינו בהמשך, אבל בינתיים מה שחשוב לנו זה החלק האחרון. אנחנו יכולים לראות שבסוף ההודעה אנחנו שולחים לו רשימה של כל מי שמחובר אם "&" ביניהם (**שימו לב כולל הוא עצמו...**). כמובן שליפני כתבנו את אורך הרשימה שלנו, כדי שהוא ידע כמה לקרוא. אבל שימו לב הפעם זה לא שני בתים, זה חמש בתים. למה? כי כאן אנחנו מצפים למחרוזת יותר גדולה... הפעם המקסימום הוא 99999, פרגנו לנו...

אוקי, קחו נשימה. עד לכאן אנחנו בשלב החיבור. עכשיו בעצם, יכולים להתחבר אלינו לשרת, ואנחנו יכולים להבין מי מחובר.

אבל לפני שממשיכים, עוד נקודה קטנה. רק בפעם הראשונה שהלקוח מתחבר, הוא יאמר לנו מי הוא (כלומר את השם שלו). אחרי זה הוא סתם שולח לנו הודעות, הוא לא יזכיר לנו מי הוא בכל הודעה. לכן, אחרי השלב של התחברות כדאי שהת'רד שמטפל בכל לקוח, ישמור גם את השם של הלקוח שהוא מדבר איתו. זה יעזור לכם בהמשך...

אוקי, יש לנו לקוחות מחוברים. מה עכשיו?

חוזרים לנעליים של הלקוח... מה הלקוח יכול לרצות עכשיו? מה יכולות להיות הבקשות שלו?

1. כמובן הוא רוצה **לשלוח הודעה**, אחרי הכל זאת המטרה של האפליקציה שלנו...
2. יכול להיות שנהייה לו משאמם והוא רוצה לדעת אם משהו ספציפי שלח לא הודעה.
3. ויכול להיות שהוא סתם רוצה לדעת אם עוד משהו התחבר, סתם הוא אוהב להישאר מעודכן.

עכשיו חשוב להוסיף, השרת שלנו לא עונה למי שלא שאל שאלה... והכוונה היא שאם משהו שולח הודעה אני שולח לו תשובה. שזה אומר שאם הלקוח רוצה להישאר מעודכן הוא פשוט שואל כל הזמן "נו? יש משהו חדש?", "תגיד אליס שלחה לי הודעה?", "אולי מישהו חדש התחבר?" , "רגע, אולי עכשיו אליס שלחה לי הודעה?" והבנתם את הרעיון...

עכשיו בא נראה מה אנחנו צריכים לענות על כל שאלה:

אם הלקוח רוצה לשלוח הודעה למשהו ספציפי, אז אני כמובן צריך לשמור את ההודעה שלו איפה שהוא ופשוט להחזיר לו אם זה הצליח או לא. אבל אני יודע שהוא חפרן, ושאחרי שנייה הוא ישאל אותי אם מישהו שלח לו הודעה חדשה. אז כמובן מקדימים תרופה למכה, ואני שולח לו גם את הצ'אט עם הלקוח האחר שאיתו הוא מדבר. אבל כמובן ששנייה אחרי זה הוא גם ירצה לדעת אם מישהו חדש התחבר. אז אני שולח לו גם את כל מי שמחובר. **אז בשביל לסכם, אני מחזיר לו צ'אט, ורשימה של כל מי שמחובר**. מיד נראה איך ההודעה הזאת נראית בפועל. אבל קודם כל, בא נמשיך לראות איזה עוד אופציות יש לנו.

אם הוא סתם רוצה לדעת אם משהו ספציפי שלח לו הודעה, אז אני יכול לענות לו כמו שעניתי להודעה הקודמת, **הצ'אט, ורשמה של כל מי שמחובר**. ככה אני מצמצם את ההודעות שאני צריך להחזיר לו.

כמובן שאם הוא רק רוצה לדעת מי מחובר, אז שייהנה. אני יענה לו בדיוק כמו שאני עונה בהתחברות.

אוקי, מגניב. אבל איך זה נראה בפועל?

אז ככה:

- **"Hi 00002 alice 05 204"** - זה אם הלקוח רוצה לשלוח הודעה ספציפית. פה לצורך העניין, בוב שולח לאליס "הי". איך אני יודע שזה בוב, כי אני חכם, ושמרתי את השם שלו.

אז שימו לב איך זה עובד. החלק הראשון, זה הקוד של ההודעה, בעצם סימן לשרת מה אנחנו רוצים לעשות. אחרי זה אנחנו אומרים למי אנחנו רוצים לשולח את ההודעה. במקרא הזה לאליס (כמובן שאנחנו גם אומרים מה הגודל של המחרוזת, שהשרת ידע כמה לקרוא. ופה אנחנו לא צריכים יותר משני בתים... כמובן שזה קבוע שמגשימים החליטו ואנחנו לא יכולים לשנות את זה אם בא לנו...). ובסוף יש לנו את ההודעה עצמה שאנחנו רוצים לשלוח לאליס (ושוב כמובן גם את גודל המחרוזת, רק שהפעם פרגנו לנו בחמישה בתים... 99999 מלאאא!)

- **"00000 alice 05 204"** - מה זה אומר אם הוא שלח לנו הודעה ריקה לאליס? יפהפייה, הוא סך הכל רוצה עדכון, ואנחנו נחזיר לו את אותה הודעה בדיוק שנחזיר במקרא שהוא כן שלח הודעה.
- **"00000 00 204"** - אז מה זה אומר? מצוין! הוא סך הכל רוצה עדכון על מי מחובר! מושלם!

עכשיו בא נראה איך התשובה שלנו נראית. אני מזכיר לכם שאנחנו צריכים להחזיר את **הצ'אט** של מי שהוא מדבר איתו, ואת **כל מי שמחובר**. אז ככה זה נראה:

"101 00002 Hi 05 alice 00009 bob&alice"

אז מה אנחנו רואים כאן? דבר ראשון מזהה ההודעה. שאצלנו אם שמתם לב זה רק 101. אחרי זה יש לנו את כל ההודעות שנשלחו בין בוב ואליס. שכרגע זה רק "הי". החלק הזה נראה טיפה אחרת אבל בשביל לא לסבך את העניינים כרגע נשאיר את זה ככה. אחרי זה אנחנו שולחים עם מי הצ'אט שלנו כרגע. שפה התשובה היא אליס, כי בוב שלח את ההודעה ולכן כמובן שהשיחה איתו. ואת החלק האחרון אתם כבר מכירים מהשלב של ההתחברות - בעצם רשימה של כל מי שמחובר.

עכשיו לפני שאנחנו ממשיכים, זוכרים שאנחנו צריכים לשמור איפה שהוא את ההודעות שהם שולחים אחד לשני? יפה, אז כל צ'אט בין שני אנשים אנחנו שומרים בקובץ. ואיך נקרא לקובץ? אם נקרא לו bob&alice.txt אז מה יקרא אם אליס היא זאת שתשלח הודעה? יצא מצב שיש לנו שני קבצים. אחד bob&alice.txt והשני alice&bob.txt, רע מאוד... אז בשביל שלא יקרא המצב הזה החלטנו על פרוטוקול. אנחנו נעשה קובץ שהשם הגדול מביניהם יהיה ראשון (מבחינה מילונית "אבא" יותר גדול מ"פלפל", כי "א" מגיעה לפני "פ"). בדוגמה שלנו הקובץ יהיה alice&bob.txt.

עכשיו אנחנו לא סתם זורקים שם את ההודעות, יש פורמט של מגשימים. למה שנשתמש בפורמט ספציפי? למה שלא נעשה אחד משלנו? כי אנחנו לא אלה שמבינים את התוכן של הקובץ. זה הרבה יותר פשוט, אנחנו שולחים פשוט את כולו ללקוח, והוא כבר מבין מה כתוב שם. אבל בשביל שהוא יבין אנחנו צריכים להקפיד על הנוסח הנכון. וזה הנוסח:

&MAGSH_MESSAGE&&Author&<author_username>&DATA&<message_data>

אם נכניס את ההודעה של בוב לקובץ, הוא אמור להראות ככה.

&MAGSH_MESSAGE&&Author&bob&DATA&Hi

שימו לב לכמה דברים. חשוב לוודא שאתם כתבתם את הכל כמו שצריך, ולא תעיתם באיזה אות או משהו כזה. בנוסף שימו לב שאם הלקוח לא מגיב יותר, הרבה פעמים זה קורה כי לא שלחתם לו את ההודעה בנוסח תקין. אז תוודאו שאתם שולחים הכל בצורה תקינה.

בשביל שתבינו עד הסוף, אם אליס ענתה חזרה לבוב, הקובץ צריך להראות ככה:

&MAGSH_MESSAGE&&Author&bob&DATA&Hi&MAGSH_MESSAGE&&Author&alice&DATA&Hi-back!

כמובן שאם ההודעה ריקה אין למה להכניס אותה לקובץ. שימו לב שאתם לא נופלים על קטנות... ועכשיו שאתם יודעים איך השרת שומר את הצ'אטים שלו. אז אני יכול להראות לכם מה באמת מחזיר השרת, וזה נראה ככה:

"101 00034 &MAGSH_MESSAGE&&Author&bob&DATA&Hi 05 alice 00009 bob&alice"

שזה בדיוק אותו דבר. רק במקום לשלוח רק "הי" שלחנו את כל התוכן של הקובץ. וכל פעם שלקוח ירצה לקבל עדכון אנחנו פשוט נשלח לו את כל התוכן של הקובץ. פשוט יותר, לא חושבים?

אוקי, בשלב הזה יש לכם מושג איך מדברים השרת והלקוח. אבל עדיין. מה? למה הUML כל כך מסובך??

אז זה באמת השלב לראות את הUML היפהפייה הזה. ובאו נחקור אותו קצת...

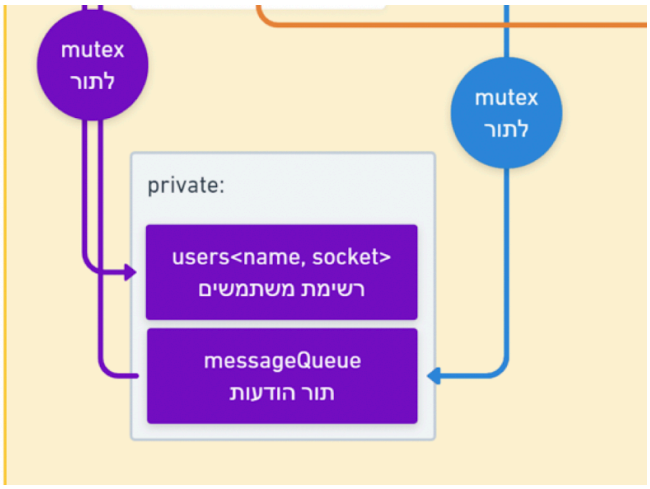
ת'רד שמקבל לקוחות יש לנו. ת'רד לכל לקוח, גם יש לנו. עד לפה סבבה. עכשיו כמובן שכל לקוח צריך להאזין להודעות. וראינו שיש לו שני אופציות להודעות שהוא יכול לקבל. הודעת 200 שזאת הודעת התחברות, והודעת 204 שזאת הודעת עדכון או שליחת הודעה למישהו. על כל אלה השרת צריך להגיב בהודעת 101 שזה הודעת העדכון שלו. אבל יש משהו מוזר בUML, **מי זה "הת'רד שאחרי לטפל בהודעות"? מה אני צריך אותו?**

שאלה מדהימה. ופה אנחנו צריכים להיזכר בכמה דברים. על כל הודעה שהלקוח שולח לי, אני שולח לו גם את רשימת המחוברים, נכון? (תגללו טיפה למעלה, תבדקו אותי...) אבל אם כל ת'רד של לקוח יתחיל להתעסק אם הרשימה של המשתמשים המחוברים, יהיה לנו data race! וכמובן שאנחנו נצטרך mutex-ים! לא כיף... אבל יותר מזה, זה לא נגמר כאן כי אנחנו כותבים גם לקבצים. מה אם הת'רד של בוב והת'רד של אליס יכתבו לאותו הקובץ ביחד בו זמנית. שוב צריך mutex-ים! אבל פה זה עוד יותר מסובך כי יש לנו מספר לא מוגדר של קבצים, אז אנחנו לא יודעים כמה mutex-ים נצטרך. בקיצור כאב ראש גדול...

אבל... מה אם לא אנחנו היינו ניגשים לרשימה של המשתמשים. מה אם לא אנחנו היינו כותבים לקבצים. אלא היינו נותנים את המשימה הזאת לת'רד אחד, שהוא יהיה אחראי על זה. והוא יעשה את הקריאה והכתיבה לקבצים. והוא יוסיף או יסיר משתמשים ככה לא יהיה לנו data race! שזה מושלם לנו!

בדיוק בגלל זה אנחנו יוצרים את הת'רד ש"אחראי לטפל בהודעות". אבל איך אנחנו מתקשרים בין ת'רדים? מצוין! משאב משוטף. בדיוק בשביל זה יש לנו תור הודעות, שאליו כל ת'רד של לקוח יכניס את ההודעה שהוא קיבל. והת'רד שאחראי לטפל בהודעות ירוץ כל פעם ויעשה את המשימות ששלחו לו בתור הודעות.

וזה מה שאנחנו רואים כאן:

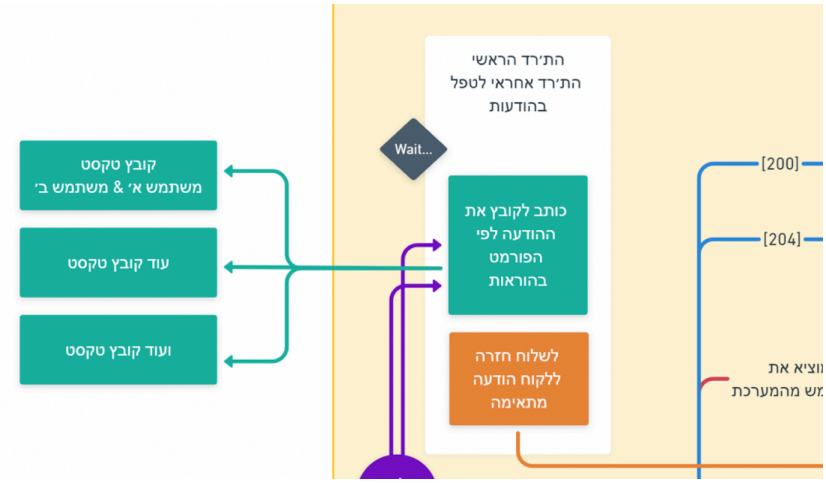


אנחנו יכולים לראות שכל ת'רד של לקוח מכניס הודעה לתור. והת'רד שמתפל בהודעות מוציא אחת אחת ומטפל בא. למה יש שם mutex? יפה כי זה משאב משוטף. אולי הורדנו הרבה משאבים משוטפים. אבל עדיין יש לנו אחד, ואנחנו צריכים לדאוג שלא יקרא data race! לכן אנחנו נועלים לפני כל נגיע בתור הודעות. גם הכנסה וגם הוצאה.

אוקי, אז אני (הת'רד שמטפל בלקוח) מחכה לקבל הודעה מהלקוח ואז אני מכניס אותה לתור הודעות. **אבל למה יש שם סוקט וכל זה?** אז זהו, אנחנו יכולים לשלוח הודעה לת'רד שמטפל בהודעות. אבל איך הוא יכול להחזיר לנו הודעות? ועוד משהו אנחנו בכל הודעה שולחים את רשימה של כל מי שמחובר, אבל רק הת'רד שמטפל בהודעות יכול לגשת לרשימת המחוברים. וגם אנחנו שולחים חזרה לפעמים את התוכן של הקובץ, שלזה גם רק לת'רד שמטפל בהודעות יש גישה. אז אנחנו חייבים למצוא דרך שהוא יתקשר איתנו חזרה, שהוא ישלח לנו את מה שאנחנו צריכים, אנחנו נשלח אותו ללקוח. **או...**

או שהוא בעצמו ישלח חזרה את ההודעה ללקוח... למה להעביר את זה דרכנו. אנחנו נשלח לו גם את הסוקט (סוקט זה רק מספר, לא?) ואז הוא יטפל בהודעה והישלח אותה חזרה ללקוח. ואנחנו (הת'רדים של הלקוחות) נמשיך לקבל הודעות.

רק שימו לב! אם חילקנו את קבלת ההודעות ושליחת ההודעות לשני ת'רדים, חשוב להיזהר. לקבל הודעות בת'רד אחד, ולשלוח באחר זה לא data race זה לא אותו משאב. וזה ממש תקין לקבל הודעה ובאותו הזמן לשלוח אחת. אבל **אסור שיקרה מצב ששני ת'רדים שולחים או מקבלים הודעות על אותו סוקט** זה כן data race. לכן חשוב להקפיד שהת'רד של הלקוחות רק מקבל הודעות (recv) והת'רד שמטפל בהודעות רק ישלח הודעות (send).



וזה בעצם מה שאנחנו רואים כאן, יש לנו את הת'רד שמטפל בהודעות. והוא קורא את ההודעות מהתור. ככה הוא יודע מה המשימה שלו ולאיזה סוקט להחזיר תשובה. ככה הוא יכול לגשת לקבצים שהוא צריך. וככה הוא יכול להוסיף משתמש ולהסיר משתמש. בלי שאף אחד יפריע לו. ובסוף הוא גם שולח הודעה חזרה ללקוח לפי הסוקט שהוא מקבל.

אז בעצם הת'רד שמטפל בלקוחות. צריך לרוץ בלולאה אין סופית ולבדוק כל פעם אם יש הודעה חדשה? כמובן שיש דרך טובה יותר, כי אין למה שהוא יפעל סתם. באו פשוט נעדכן אותו כל פעם שאנחנו מכניסים הודעה חדשה ככה הוא לא יצטרך לרוץ סתם! condition_variable! נכון! בדיוק בשביל זה!

אנחנו נעשה wait בצד של הת'רד שמטפל בהודעות. וכל פעם שת'רד של לקוח מכניס משימה, הוא יעשה notifiy_one ככה הת'רד שמטפל בהודעות התעורר ויטפל בהודעה.

כדאי ליצור אובייקט שייצג משימה. ולא להעביר פשוט מחרוזת. תחשבו על זה שת'רד שמטפל בהודעות צריך לטפל בהודעות של כולם. לכן עדיף לעשות את כל הבדיקות שרק אפשר אצל הת'רדים של הלקוחות ולשלוח לו תמצית של המשימה. אובייקט שיהיה לו username, socket, message וכו'. אבל זה לבחירתכם. תעשו מה שאתם חושבים לנכון...

עוד משהו שלא דיברנו עליו והוא חשוב. מגשימים יצרו לכם קובץ helper שהוא ממש עוזר הוא מרכיב בשבילכם את כל ההודעות בפרוטוקול. גם השליחה וגם הקבלה. אם לא הבנתם מה הוא עושה, תשאלו! סתם יקרא בסוף מצב שאתם כותבים קוד שכבר כתבו בשבילכם וחבל...

וואו, כן, זה וואחד פרויקט. אבל הוא מגניב!

לסיכום אנחנו יוצרים לכל לקוח שהתחבר ת'רד שמטפל בו. הת'רד הזה מאזין לבקשות של אותו לקוח. הבקשה הראשונה שלו חייבת להיות התחברות, אם לא - סגרו לו את הסוקט! אנחנו לא אוהבים אותו... מיד אחרי זה אנחנו צריכים להאזין לנצח (או עד שהלקוח התנתק) לבקשות של עדכון או שליחת הודעה. שהם בעצם אותה הודעה. 204 ליתר דיוק. היא יכולה להיות קצת ריקה, ואנחנו נבין שהוא רק רוצה עדכון...

אנחנו נוודא שההודעה תקינה והוא לא שלח משהו מוזר. **ואם ההודעה תקינה, אנחנו נדחוף אותה לתור הודעות** (כמובן שנגעל את הmutex לפני, ושנעשה לו notifiy אחרי) ונחזור להאזין להודעות.

חשוב להוסיף אם הלקוח התנתק בפתאומיות אתם תקבלו 0 מהפונקציה שמחזירה את הקוד-הודעה (Helper::getMessageTypeCode) ככה תוכלו לדעת שקרתה תקלה ותנתקו את הלקוח הזה מהשרת (וכמובן שלחו לת'רד שמטפל בהודעות הודעה שהתנתק הלקוח...).

באותו הזמן הלקוח שמטפל בהודעות. רץ בלולאה אין סופית, ומחכה כל פעם לעדכון אם יש לו מה לעשות. כשיש לו הודעה חדשה, הוא מטפל בה ושולח חזרה את ההודעה - 101. אלא בעצם הדברים שיש לו לעשות:

א' - **להוסיף לקוח חדש**. שזאת בעצם התחברות.

ב' - **לשלוח עדכן על צ'אט מסוים, ואם יש הודעה גם להוסיף אותה לקובץ** (בעצם אם יש הודעה הוא יכתוב לקובץ ואם אין הוא רק יקרא ממנו את כל הצ'אט וישלח ללקוח...). שימו לב שאם הוא רק רוצה עדכון למי מחובר אז אין פשוט קובץ לקרוא ממנו, אתם יכולים לעשות שאם הוא לא מצליח לפתוח את הקובץ הוא יחזיר "" בתוכן של הקובץ. ככה הלקוח יקבל רק את הרשימה של מי שמחובר...

ג' - **לנתק לקוח**. שזה בעצם להוציא אותו מהרשמת לקוחות מחוברים. שימו לב שלא צריך לשלוח ללקוח הודעה כל שהיא, כי הוא כבר מנותק... אין למי לשלוח...

זהו! אם יש לכם את זה, יש לכם שרת! כמובן שזה לא כזה פשוט. אבל זה גם פחות מפלצת עכשיו. אז תיהנו!

אם יש שאלות בשביל מה יש יהל? כן כן כמובן, גם אני תמיד פה בשבילכם...