הפורט הזה. דבר זה יהיה נכון גם, למשל, אם תנסו להריץ שוב את הסקריפט שלכם לאחר שהוא קרס. על מנת להתגבר על בעיה זו, תוכלו לשנות את הפורט בו אתם משתמשים. זכרו לשנות את מספר הפורט גם בצד הלקוח וגם בצד השרת.

בנוסף, לאחר שמתבצעת קריאה ל-**close** עשוי לעבור זמן מסוים עד שמערכת ההפעלה באמת תשחרר את הפורט. ייתכן שתיתקלו בכך במהלך העבודה על התרגיל.

כעת אתם מצוידים בכל הידע הדרוש לכם על מנת לפתור את התרגיל. בהצלחה!

תרגיל 2.6 – שרת פקודות בסיסי

בתרגיל הראשון התבקשתם לכתוב לקוח, ובתרגיל השני התבקשתם לכתוב שרת. בתרגיל זה עליכם לכתוב הן את השרת, והן את הלקוח. חלק מהאתגר בתרגיל הינו כתיבת פרוטוקול למשלוח הודעות בין השרת והלקוח.

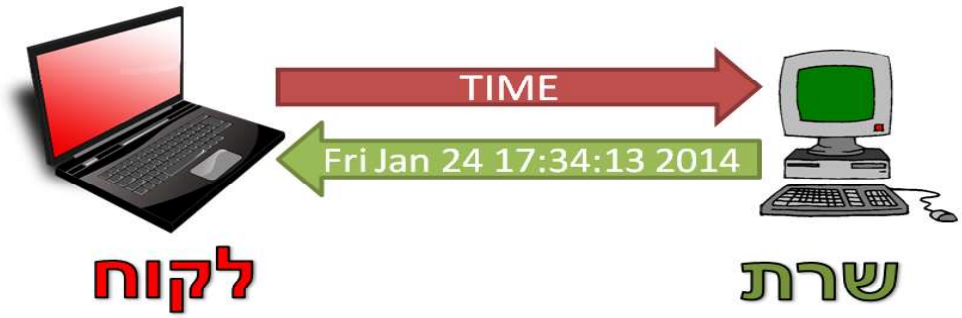
<u>שלב 1:</u>

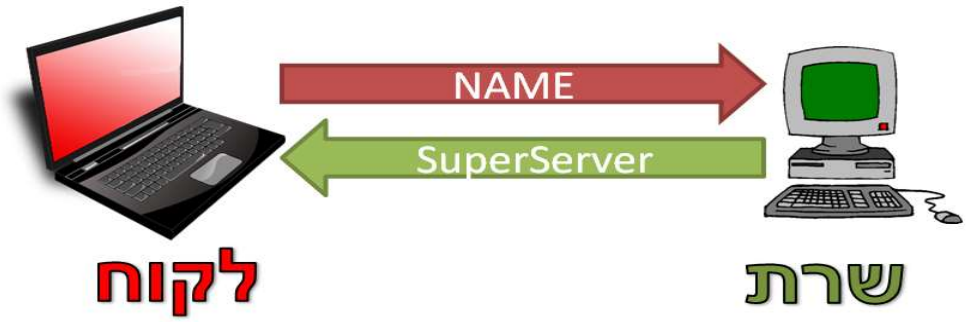
עליכם לכתוב מערכת שרת-לקוח, כאשר השרת מבצע פקודות שהלקוח שולח אליו, ומחזיר ללקוח תשובה בהתאם. על כל בקשה של הלקוח להיות באורך של ארבעה בתים בדיוק. אורך התגובה יכול להיות שונה בהתאם לבקשה.

להלן רשימת הבקשות שיש לתמוך בהן:

- TIME – בקשת הזמן הנוכחי. על השרת להגיב עם מחרוזת שכוללת את השעה הנוכחית אצלו.
  - היעזרו במודול time שמובנה בפייתון.
- NAME – בקשת שם השרת. על השרת להגיב עם מחרוזת שמייצגת את שמו. השם יכול להיות כל מחרוזת שתבחרו.
- RAND – בקשת מספר רנדומלי. על השרת להגיב עם מספר רנדומלי שנע בין הערכים 1 ל-10.
  - היעזרו במודול random המובנה בפייתון.
- EXIT – בקשת ניתוק. על השרת לנתק את החיבור עם הלקוח.

להלן דוגמאות לתקשורת:





שלב 2:

תרגול כתיבת פרוטוקול – האם בשלב 1 הלקוח שלכם ביצע socket.recv(1024)? כעת, צרו פרוטוקול שמאפשר לשלוח מהשרת ללקוח הודעות באורך שונה. יכולת זו תשמש אתכם בתרגילים בהם לא תוכלו להניח שאורך ההודעה מהשרת ללקוח הוא 1024 בתים או מספר קבוע כלשהו – לדוגמה בהעברת קובץ.

אפשרויות לדוגמה:

- השרת ישלח ללקוח הודעה שכתוב בה מה אורך המידע שעליו לקבל בתור תשובה.
- השרת ישלח ללקוח הודעה מיוחדת שמשמעותה "שליחת המידע הסתיימה".

שלב 3:

כיוון שאנו עוסקים בהגנת סייבר, עלינו לכתוב שרת יציב, כלומר גם אם הלקוח שולח "זבל", לשרת שלנו אסור לקרוס. בדקו את יציבות השרת באמצעות הודעות מסוגים שונים.

הנחיות לתרגיל 2.6

לפני שאתם ניגשים לכתוב את הקוד, בצעו תכנון ראשוני. חשבו מה בדיוק תממשו בצד הלקוח ומה בצד השרת, ונסו לצפות מראש בעיות בהן תתקלו.

בצד הלקוח – ראשית, עליכם לבקש מהמשתמש לבחור באחת מהפקודות שצוינו לעיל (רמז: היעזרו בפונקציה **raw_input**).

לאחר מכן, שלחו את הבקשה לשרת, קבלו את התשובה והציגו אותה למשתמש.

בצד השרת – עליכם לקבל חיבור מהלקוח, להבין את הבקשה שלו ולהגיב אליה בהתאם. לאחר מכן, נתקו את החיבור ותוכלו לספק שירות ללקוח חדש.

בהצלחה!

תרגיל 2.7 – שרת פקודות מתקדם- טכנאי מרוחק (אתגר)

עד כה בניתם שרתים ולקוחות שתקשרו עם זה עם זה וביצעו פעולות פשוטות. בתרגיל זה עליכם לתכנן מערכת שרת-לקוח, ולאחר מכן לממש אותה. המערכת תאפשר לטכנאי לתקשר עם מחשב