

הרעיון בתקשורת בין מחשבים הוא לנהל דו-שיח על פי פרוטוקול מוסכם. כדי לדעת עם מי, יש לנו את הסוקט שנוצר בעזרת כתובת IP ופורט משני הצדדים. המטרה לנסות ולבנות פרוטוקול 'עמיד' בפני זעזועים (בנוסף לתגובות המצופות).

### קליינט

**התוכנית הראשית** קוראת לפונקציה שמבצעת **קישור ראשוני** `initial_contact` (לחיצת-יד) ע"י יצירת אובייקט מסוג סוקט-קליינט וקישור (`connect`) ל-`ip/port`. אם הקישור עבד, משלוח של הודעה סתמית ("Hi") בעזרת `send`, ואח"כ בציפיה להודעה מוחזרת מבצעים: `recv` באורך 48 (כך הגדרנו בשרת את התשובה). שימו לב `send` ומייד `recv`.

נא לזכור ש-`recv` היא פקודה מסוג `blocking`, כלומר הקליינט יחכה לתשובה ויהיה 'תקוע' עליה (כמו `raw_input`).

**פונקציית הקישור הראשוני** גם תדפיס את ההודעה שהתקבלה מהשרת ותחזיר אם הכל תקין `True`, את מה ששלח השרת ואת אובייקט הסוקט (להמשך השיחה). אם הייתה בעיה כלשהי, היא תחזיר אינדיקציה שהייתה בעיה (`False`).

התוכנית הראשית, לאחר הקישור הראשוני, אם הכל תקין, תקרא לפונקציה של משלוח בקשות `send_requests` (האופציות שהגדרנו) עם התשובה האחרונה של השרת וסוקט של הקליינט. אם הייתה בעיה, התוכנית תסתיים.

**פונקציית משלוח בקשות** תיכנס ללולאה, כל עוד ישנה תשובה מהשרת. תבקש קלט הממשתמש (בקשה). הקלט סיכמנו שיהיה באורך 4 (לכן נתעלם מהיתר) ונשלח אותו (באותיות גדולות כמוסכם) לשרת. מייד לאחר מכן נחכה לתשובת השרת באורך 4 בתים - מה שיכיל את אורך התשובה האמיתית (`recv(4)`), על פי אורך זה נחכה להמשך תשובת השרת באורך המצופה (`recv(messge_length)`) ואותה נדפיס. על פי הפרוטוקול שפותח, קבענו שהשרת תמיד יצרף 4 בתים ראשונים לתגובתו שיכילו את אורך התגובה, לתגובה עצמה. אם השרת לא שולח יותר תשובות, מסתיימת לולאת הבקשות וגם הפונקציה.

שרת

**התוכנית הראשית** קוראת להאזנה ראשונית (`initial_listen`) שקושרת את הסוקט לשרת (כתובת ip ופורט) `bind, listen` ומחזירה אובייקט זה (להמשך השיחה). לאחר מכן לולאה אינסופית של בדיקה אם הקליינט שלח בקשה ראשונה (`check_client`). כלומר לולאה זו רצה וכל פעם 'מחפשת קליינט', עד אשר נוריד את השרת בעצמנו.

**פונקציה בדיקת הקליינט** (`check_client`) מקבלת את הסוקט של השרת (שחזר מ `initial_listen`) ובעזרתו מבצעת `accept` לקבלת הסוקט של הקליינט והצמד `ip,port`. על פי הסוקט שהתקבל מה- `accept` השרת יבצע `recv` עבור קבלת לחיצת היד מהקליינט. ואז השרת ישלח ללקוח את ההודעה הראשונית שלו, שתפרט מה הבקשות שהקליינט יכול לבקש (באורך 48). לאחר מכן השרת יבצע `recv(4)` לקבל את בקשת הקליינט. הפונקציה תחזיר את הסוקט של הקליינט ואת הבקשה.

התוכנית הראשית של השרת תקרא לפונקצית בקשות מהשרת (`serve_requests`) שתהיה מוכנה לקבל כל בקשה מאותו קליינט ללא סוף.

**פונקצית הבקשות** מהשרת מקבלת את הבקשה מהקליינט וסוקט הקליינט. ואז מריצה לולאה אינסופית שבה בכל סיבוב על פי סוג הבקשה היא בונה את התגובה חזרה, שולחת אותה (`send`) ומחכה לבקשה הבאה (`recv`). הלולאה תסתיים על ידי `break` במידה והקליינט ביקש `EXIT` (ואז נסגור את הסוקט שלו) ונהיה מוכנים לקליינט חדש.