



רשתות תקשורת מחשבים
סמסטר א' תשפ"ו

תרגיל בית מס' 1 להגשה עד 23/11/2025

חלק ראשון (85%) – Client-Server Program

בתרגיל זה נכתוב תוכנת שרת-לקוח בסיסית בה לקוחות יוכלו להשתמש בשרת לצורך ביצוע מספר פעולות. השרת והלקוח ישוחחו בפרוטוקול אפליקציה אשר תגדירו בעצמכם, כאשר התעבורה תתבצע על גבי TCP או UDP, לבחירתכם.

מטרות התרגיל:

- שימוש נכון ב-socket programming, עבודה מול ממשקי רשת.
- תכנון ופיתוח פרוטוקול תקשורת ברמת האפליקציה.
- תמיכה בריבוי לקוחות במקביל.

השרת:

אתחול השרת מתבצע בשורת הפקודה (ולאחריה השרת ממתיין ומשרת לקוחות).

`./ex1_server.py users_file [port]`

כאשר:

- `users_file` הוא path לקובץ טקסטואלי tab-delimited שמכיל שתי עמודות: שם משתמש וסיסמא (עבור אותו המשתמש). למשל:

Bob	simplepass
Alice	BetT3RpAas

כאשר בין שני השדות יש `tab`. ניתן להניח שאין מוסיפים לקוחות בעת ריצת השרת.

- הפורט להאזנה (`port`) הוא פרמטר אופציונלי עם ערך ברירת מחדל 1337.

לשם פשטות, מרגע שהשרת מתחיל לרוץ הוא אינו מסיים את ריצתו. השרת יוכל לשרת מספר לקוחות במקביל וירבב ביניהם בעזרת שימוש ב- `select` כפי שנראה בתרגול. כלומר, בכל רגע ייתכנו מספר לקוחות המחוברים לשרת במקביל, והוא משרת את כולם. באותו הזמן, השרת מאזין ויכול לקבל לקוחות חדשים שמתחברים אליו. השימוש ב- `select` יאפשר לשרת להימנע מקריאות חוסמות, ללא שימוש ב- `multi-threading`.

הלקוח:

אתחול הלקוח מתבצע בשורת הפקודה:

`./ex1_client.py [hostname] [port]`

כאשר `hostname` ו-`port` הם פרמטרים אופציונליים שערכי ברירת המחדל שלהם הינם:

`hostname = localhost` •

`port = 1337` •

שימו לב, לא ניתן לספק `port` ללא `hostname`, אך ניתן לספק `hostname` ללא פורט (במקרה כזה רק ה-`port` ישתמש בערך ברירת המחדל). ה-`hostname` יכול להיות שם או כתובת IP.



רשתות תקשורת מחשבים
סמסטר א' תשפ"ו

התחברות:

כשלקוח חדש מתחבר לשרת, השרת שולח לו ברכת שלום:

Welcome! Please log in

אחרי שהלקוח מקבל את ברכת השלום מהשרת, הוא מזין את פרטיו בפורמט הבא:

User: username_of_user

Password: password_of_user

הפרטים יישלחו לשרת. אם הם נכונים, השרת מקבל את הלקוח, מציג הודעה הכוללת את שם הלקוח:

Hi {username_of_user}, good to see you

אחרת יודפס:

Failed to login.

ותינתן לו האפשרות להזין שוב את פרטיו.

פקודות נתמכות:

1. שליחת מחרוזת סוגריים, שתראה כך:

parentheses: X

כאשר X מחרוזת סוגריים (לדוגמה: (())()). השרת יבדוק אם מדובר במחרוזת סוגריים מאוזנת (כל סוג שנפתח גם נסגר) ויחזיר את התשובה, שתודפס כך:

the parentheses are balanced: yes/no

2. מציאת הכפולה המשותפת המינימלית עבור שני מספרים. הפקודה תראה כך:

lcm: X Y

כאשר X ו-Y הם שני מספרים שלמים (signed int).

השרת יחשב את הכפולה המשותפת המינימלית עבור שני המספרים ויחזיר את התשובה ללקוח, שידפיס אותה כך:

the lcm is: R

לשם פשטות – ניתן להניח כאן כי גודל הפרמטרים (X ו-Y) הוא עד גודל int.

3. צופן קיסר. הפקודה תראה כך:

caesar: plaintext X

כאשר plaintext הינה מחרוזת (שיכולה להיות מורכבת ממספר מילים) ו-X הינו מספר שלם. השרת יחזיר את התשובה ללקוח, שידפיס אותה כך:

the ciphertext is: Y

כאשר Y הינה המחרוזת (באותיות אנגלית קטנות) המתקבלת מהפעלת צופן קיסר על מחרוזת plaintext עם המפתח X (כלומר, ההצפנה "מזיזה" כל אות במחרוזת ב-X אותיות באלפבית).

במידה והמחרוזת מכילה תו שאינו אות באנגלית (קטנה/גדולה) או רווח, על השרת להחזיר:

error: invalid input

4. התנתקות, שתראה כך (ללא פרמטרים נוספים):



רשתות תקשורת מחשבים
סמסטר א' תשפ"ו

quit

לנוחיותכם, דוגמת הרצה מצורפת בסוף הקובץ.

בכל שלב, במידה והלקוח לא עבר את שלב ההתאמתות ויישלח פקודה אחרת לשרת, השרת יסגור את החיבור מולו. כמו כן לשם פשטות ניתן להניח כי אם מתקבל קלט לא בפורמט צפוי מהמשתמש, התוכנה תדפיס הודעת שגיאה ותתנתק מהשרת.

דרישות התרגיל:

עליכם לממש את השרת והלקוח כך שיממשו את הפקודות והפעולות שהוגדרו בתרגיל. לשם כך, תתכננו פרוטוקול אפליקציה מתאים שיעבוד על גבי פרוטוקול TCP או UDP, וממשו אותו. את פרוטוקול האפליקציה יש **לתעד** בבירור, באופן שאמור לאפשר לכל אדם לממש לקוח או שרת ש"ידברו" עם התוכנות שהגשתם. הגישו בנוסף קובץ Readme שבו תיאור של הפרוטוקול שבחרתם לממש, והסבר לבחירתכם במימוש TCP/UDP.

הדגש בבדיקת הקוד שתגישו יינתן כמובן לעבודה מול ה-socket-ים בתוכנה (שימוש ב-API, בדיקת שגיאות, וכו') ועל מימוש התקשורת בין השרת והלקוח כך שיהיה יעיל והגיוני בהתאם לדגשים עליהם דיברנו בכיתה.



רשתות תקשורת מחשבים
סמסטר א' תשפ"ו

חלק שני (15%) – עבודה עם Wireshark

הורידו את תוכנת wireshark מהאתר <https://www.wireshark.org>. תוכנה זאת מאפשרת לצפות בפקטות שהתקבלו למחשב שלכם דרך אחד מכרטיסי הרשת הקיימים אצלכם, כפי שראינו בתרגול.

הריצו את התוכנה שכתבתם, בחרו את כרטיס הרשת הרלוונטי, השתמשו ב-filter מתאים, והראו אלו פקטות עוברות בין הלקוח והשרת כאשר הלקוח מתחבר לשרת. צרפו קובץ המכיל תעבורה רלוונטית של הסשן שבו רואים את התעבורה הרלוונטית (יש ב-wireshark אפשרות של save as, בפורמט pcap).

כמו כן, בחרו חלק בתהליך התקשורת ביניהם לבחירתכם (לדוגמא - התחברות, או שליחת שאלתא מהלקוח, או קבלת תשובה מהשרת, וכו'). וצרפו צילום מסך מתאים שמראה אותו ב-wireshark יחד עם תיאור קצר של התוכן/פרוטוקול אפליקציה שלכם שמועבר שם.

הנחיות הגשה:

הגשת התרגילים תתבצע דרך המודל.

הגישו קובץ ZIP יחיד ששמו:

EX1_ID1_ID2

או (כתלות בעבודה בזוגות/לבד):

EX1_ID

במידה והכנתם את התרגיל בזוג, רק אחד נדרש להגיש את התרגיל במודל.

קובץ ה-ZIP יכיל את כל קבצי הקוד, כך שיוכלו לרוץ בסביבת ה-nova בהצלחה.

ניתן לחלק את הקוד לקבצים נוספים בנוסף לקבצי ההרצה הראשיים, לפי שיקול דעתכם.

קבצי ההרצה יכולים להיות עם הסיומת .py או בלעדיה. רצוי להוסיף

```
#!/usr/bin/python3
```

.בראשית הקבצים.

קובץ Readme.pdf שיכיל תיאור של הפרוטוקול ותכנון השרת והלקוח.

הקבצים הרלוונטיים לחלק השני (העבודה מול wireshark) ובהם קובץ pcap, ותמונות/PDF עם התעבורה הרלוונטית והסבר קצר.

😊 **בהצלחה**



רשתות תקשורת מחשבים
סמסטר א' תשפ"ו

דוגמת הרצה: (הטקסט בכחול – קלט מהמשתמש).

בצד השרת:

```
./ex1_server.py ~/my_dir/users_file.txt
```

בצד הלקוח:

```
./ex1_client.py
```

Welcome! Please log in.

User: Bob

Password: simplepass

.Hi Bob, good to see you

caesar: Hello! 2

error: invalid input

כעת יתחבר לקוח נוסף:

```
./ex1_client.py
```

Welcome! Please log in.

User: Alice

Password: BetT3RpAas

.Hi Alice, good to see you

parenthese: ((())) (())

the parentheses are balanced: yes

lcm: 6 21

the lcm is: 42

quit

בוב ימשיך בפעילות במקביל:

caesar: Hello 2

the ciphertext is: jgnnq



רשתות תקשורת מחשבים
סמסטר א' תשפ"ו

[quit](#)