



רשותות תקשורת מחשבים
סמסטר א' תשפ"ז

תרגיל בית מס' 1 להגשה עד 23/11/2025

חלק ראשון (85%) – Client-Server Program

בתרגיל זה נכתבת תוכנת שרת-לקוח בסיסית בה ללקוחות יוכל להשתמש בשרת לצורך ביצוע מספר פעולות. השרת והלקוח יושחוו ב프וטוקול אפליקציה אשר תגידו בעצמכם, כאשר התעבורה תבוצע על גבי TCP או UDP.

לבחירהיכם.

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 ממתין ושרות לקוחות).

`./ex1_server.py users_file [port]`

כאשר:

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

Bob	simplepass
Alice	BetT3RpAas

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

- הפורט להזנה (port) הוא פרמטר אופציונלי עם ערך ברירת מהדול 1337.

לשם פשוט, מרגע שהשרת מתחילה לזרז הוא אינו מסיים את ריצתו. השרת יוכל לשרת לקוחות במקביל וירכב בינהם בעזרת select – כפי שנראה בתרגול. כמובן, בכל רגע יתכונו מספר לקוחות המוחברים לשרת במקביל, והוא משרת את כלם. באותו הזמן, השרת מאזין ויכול לקבל לקוחות חדשים שמתחברים אליו. השימוש ב-select יאפשר לשרת להימנע מקרייאות חסומות, לא שימוש ב-multi-threading.

הלקוח:

אתחול הלקוח מתבצע בשורת הפקודה:

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

כאשר hostname ו-port הם פרמטרים אופציונליים שערבי בברירת המחדל שלהם הינם:

- hostname = localhost
- port = 1337

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



התחברות:

כשלוקה חדש מתחבר לשרת, השרת שולח לו ברכבת שלום:

.Welcome! Please log in

אחרי שהלקה מקבל את ברכבת 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 המלוכה, מציג הודעה הכלולת את שם הלקוה:

.Hi {username_of_user}, good to see you

אחרת יודפס:

Failed to login.

ותינתן לו האפשרות להזין שוב את פרטיו.

פקודות נOMICOT:

1. שליחת מהרוות סוגרים, שתראה כך:

parentheses: X

כאשר X מהרוות סוגרים (לדוגמא: ()()) . השרת יבדוק אם מדובר ב מהרוות סוגרים מאוזנת (כל סוג

שנפתח גם נסגר) ויזיר את התשובה, שיתודפס כך:

the parentheses are balanced: yes/no

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

lcm: X Y

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

השרת יחשב את הכפולת המשותפת המינימלית עבור שני המספרים ויזיר את התשובה ללוקה, שידפיס אותה כך:

the lcm is: R

לשם פשוט – 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 מהרוות (שיכולת להיות מורכבת מספר מילים) ו-X הינו מספר שלם.

השרת ייזיר את התשובה ללוקה, שידפיס אותה כך:

the ciphertext is: Y

כאשר Y הינה המרוות (באותיות אנגלית קטנות) המתקבלת מהפעלת צופן קיסר על מהרוות plaintext עם המפתח X (כלומר, ההצפנה "מזהה" כל אות ב מהרוות ב-Xאותיות באلفבית).

במידה והמהרוות מכילה تو שאינו אות באנגלית (קטנה/גדולה) או רווח, על השרת להיזיר:
error: invalid input

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



רשותות תקשורת מחשבים
סמסטר א' תשפ"ז

quit

לנוחיותכם, דוגמת הרצה מצורפת בסוף הקובץ.

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

דרישות התרגיל:

עליכם למש את השרת והלקוח כך שיימשו את הפקודות והפעולות שהוגדרו בתרגיל. לשם כך, תחכנונו פרוטוקול אפליקציה מתאים שיעבוד על גבי פרוטוקול TCP או UDP, וממשו אותו. את פרוטוקול האפליקציה יש **لتעד** בבירור, באופן שאמור לאפשר לכל אדם למש לekoao או שרת ש"ידברו" עם התוכנות שהgasht. הגישו בנוסף קובץ **Readme** שבו תיאור של הפרוטוקול שבחרתם למש, והסביר לבחורהכם במימוש TCP/UDP.

הדגש בבדיקה הקוד שתגישיו יינתן כמובן לעובדה מול ה-socket-ים בתוכנה (שימוש ב-API, בדיקת שגיאות, וכו') ועל מימוש התקשרות בין השרת והלקוח כך יהיה יעל והגיוני בהתאם לדגשים עליהם דיברנו בכיתה.



רשותות תקשורת מחשבים
סמסטר א' תשפ"ז

חלק שני (15%) – עבודה עם Wireshark

הורידו את תוכנת wireshark מהאתר <https://www.wireshark.org>. תוכנה זאת מאפשרת לצפות בפקודות שהתקבלו למחשב שלכם דרך אחד מכרטיסי הרשות הקיימים אצלם, כפי שראינו בתרגול.
הristolו את התוכנה שכחบทם, בחרו את כרטיס הרשות הרלוונטי, השתמשו ב-filter מותאם, והראו ALSO פקודות עוברות בין הלוקה והשרות כאשר הלוקה מתחבר לשרת. צרפו קובץ המכיל תעבורת רלוונטית של הסשן שבו רואים את התעבורה הרלוונטית (יש ב-wireshark אפשרות של save as, בפורמט pcap).
כמו כן, בחרו חלק בתחום התקשרות בינהם לבחרתכם (לדוגמא - התחברות, או שילוח שאילתת מהלקות, או קבלת תשובה מהשרות, וכו'). וצרפו צילום מסך מותאם שמרת אותה ב-wireshark יחד עם תיאור קצר של התוכן/פרוטוקול אפליקציה שלכם שמועבר שם.

הנחיות הגשה:

הגשת התרגילים מתבצע דרך המודול.

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 במודול.
קובץ ZIP יכולאת כל קבצי הקוד, כך שיוכלו לרוץ בסביבת ה-nova בהצלחה.
ניתן לחלק את הקוד לקבצים נוספים לביצוי הרצאה הראשיים, לפי שיקול דעתכם.
קבצי הרצאה יכולים להיות עם הסיומת .py או בלבד. רצוי להוסיף

`#!/usr/bin/python3`

בראשית הקבצים.

קובץ Readme.pdf שיכיל תיאור של הפרוטוקול ותוכנו השרות והלוקה.

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

בהצלחה ☺



רשתות תקשורת מחשבים
סמסטר א' תשפ"ז

דוגמת הרצאה: (הטקסט בכחול – קלט מהמשתמש).

בצד השרת:

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

בצד הלקואה:

```
./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 לקואה נוספת:

```
./ex1_client.py
```

Welcome! Please log in.

User: Alice

Password: BetT3RpAas

.Hi Alice, good to see you

parentheses: ((0))0(0)

the parentheses are balanced: yes

lcm: 6 21

the lcm is: 42

quit

בוּב ימשיך בפעולות במקביל:

caesar: Hello 2

the ciphertext is: jgnnq



רשתות תקשורת מחשבים
סמסטר א' תשפ"ז

[quit](#)