

# משימת תכנות 1

## מבוא לתקשורת מחשבים

22.2.22

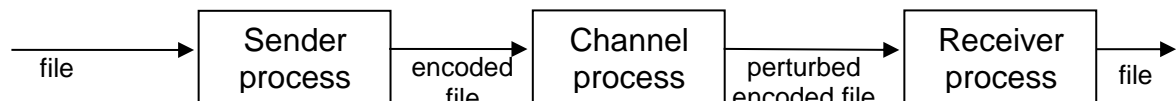
פרופ' פת-שמיר

מטרת התרגיל לרכוש הכרות בסיסית בתכנות מנגנוני השקעים (sockets) בסביבת העבודה שלכם, ולתרגל קוד המינג. התרגיל יוגש בזוגות.

**הגשה:** 23/3/2022, במידל. ראו הוראות מפורטות בנספח. נקודות יורדו על הגשה שאינה על פי ההנחיות.

### מבוא

בתרגיל זה תכתבו מערכת פשוטה בה לקוח שולח (sender) שולח הודעות אל שרת מקבל (receiver), דרך ערוץ רועש. הערוץ ממומש אף הוא ע"י שרת (ראו דיאגרמה).



### משימה בסיסית

עליכם להגיש את קוד המקור אשר יצור שלוש תכניות בנות-ביצוע (executable programs), שלושותן console applications.

### השולח

מקבל כפרמטרים בשורת הפקודה (command line arguments) את כתובת ה-IP ומספר הפורט אליו מחובר שקע הערוץ המיועד לשולח (ראו להלן). השולח יוצר שקע TCP, מתחבר דרכו לשרת הערוץ, ואז מבקש מודיע למשתמש שהוא מוכן. המשתמש מקליד שם קובץ, אשר נשלח לערוץ מקודד בקוד המינג (31,26,3). כלומר, יש לקרוא את הקובץ כסדרת בלוקים של 26 ביטים, ולשלוח כסדרת בלוקים של 31 ביטים. אפשר להניח שמספר הבתים בקובץ הוא כפולה שלמה של 26.

לאחר סיום השידור, השולח סוגר את השקע שלו, ומדווח על מספר הבתים שנקראו, מספר הבתים שנשלחו. השולח אז מתחיל שוב: פותח שקע חדש, מתחבר דרכו שוב לערוץ, ומבקש מהמשתמש שם קובץ נוסף או "quit". לאחר קבלת quit יש לסיים את הריצה.

תעתיק ריצה (execution transcript) טיפוסי של השולח עשוי להראות כך:

```
> my_sender 132.66.16.33 6342
enter file name:
myfile.txt
file length: 2600 bytes
sent: 3100 bytes
enter file name:
quit
>
```

❖ **בנוסף:** מימוש לכל אורך קובץ.

### המקבל

המקבל מקבל כפרמטרים את כתובת ה-IP ומספר הפורט אליו מחובר שקע הערוץ המיועד למקבל (ראו להלן). המקבל יוצר שקע TCP, מתחבר דרכו לפורט הערוץ שקיבל כארגומנט, מבקש שם קובץ בו ייכתוב, וממתין לקרוא. כאשר מגיע קלט על השקע (מהערוץ), המקבל מבצע את הפעולות הבאות:

- קורא את הנתונים המגיעים מהקו, מפענח אותם תחת קוד המינג, כותב את הקובץ המפוענח תחת השם שהמשתמש מזין. כמו כן המקבל שומר סטטיסטיקות:
- כאשר השקע נסגר (ע"י הערוץ) המקבל סוגר את הקובץ אותו הוא כותב, ומדווח למשתמש על מספר הבתים שנקראו, מספר הבתים שנכתבו, ומספר השגיאות שתוקנו.

- לאחר מכן המקבל מבקש שם קובץ חדש או "quit". אם ממשיכים, המקבל פותח שקע חדש לערוץ וחוזרים על הפעולות כמקודם.

תעתיק אפשרי:

```
> my_receiver 132.66.16.33 6343
enter file name:
rec_file.txt
received: 3100 bytes
wrote: 2600 bytes
corrected 12 errors
enter file name:
quit
>
```

## הערוץ

מקבל בשורת הפקודה (command line arguments) את שיטת הרעש שהוא מחדיר. השיטה היא אחת מהאפשרויות הבאות:

- אקראי: כאן כל ביט יתהפך בהסתברות נתונה באופן בלתי תלוי. אפשרות זאת תסומן ע"י הדגל "-r".
- דושיים שני פרמטרים: הראשון הוא ההסתברות לשגיאה בביט, והשני הוא זרע למחולל האקראיות (random seed). ההסתברות לשגיאה תינתן ביחידות של  $2^{-16}$  (כלומר אם הערך כאן  $n$  אז ההסתברות היא  $n/2^{16}$ ).
- דטרמיניסטי: במקרה זה יש להפוך כל ביט  $n$ -י בדיוק. יופיע הדגל "-d" והפרמטר היחיד הוא  $n$ , אורך המחזור בביטים.

כאשר הערוץ עולה, הוא פותח שקע  $S_s$  המיועד לשולח ושקע  $S_r$  המיועד למקבל, ומודיע למשתמש (בכתיבה לקובץ stdout) על כתובות ה-IP ומספרי הפורטים של  $S_r$  ו- $S_s$ . הערוץ ממתיין (ב-accept) להתחברות השולח והמקבל ל- $S_s$  ו- $S_r$ . לאחר ההתחברות של השולח והמקבל, הערוץ מחובר אליהם בשקעים חדשים  $S'_s$  ו- $S'_r$ , בהתאמה. הערוץ ממתיין על  $S'_s$  לקלט שיגיע מהשולח. כאשר מגיע קלט מהשולח, הערוץ משבש אותו לפי השיטה שנבחרה, וכותב את המידע המשובש בשקע (החדש)  $S'_r$  המחובר למקבל. כאשר נסגר שקע  $S'_s$ , הערוץ מסיים לשלוח למקבל ואז סוגר את השקע  $S'_r$ . פרט לכך, הערוץ מדווח על מספר הביטים שהעכיר ומספר הביטים שהפך ושואל האם להמשיך. תעתיק אפשרי:

```
> my_channel -d 112
sender socket: IP address = 132.66.16.33 port = 6342
receiver socket: IP address = 132.66.16.33 port = 6343
retransmitted 3100 bytes, flipped 221 bits
continue? (yes/no)
no
> my_channel -r 650 34853
sender socket: IP address = 132.66.16.33 port = 6344
receiver socket: IP address = 132.66.16.33 port = 6345
retransmitted 3100 bytes, flipped 197 bits
continue? (yes/no)
no
>
```

❖ בונוס: מימוש יעיל של פעולת הערוץ. בפרט, מימוש שאינו מבצע הגרלה עבור כל ביט בהודעה.

## הערות:

- אפשר להשתמש ב-`sscanf()` כדי לקרוא מספר ממחרזות.
- מומלץ לפתח לפי הסדר הבא: ראשית, רק שלד התקשורת, ללא קידוד וללא רעש: שולח מול ערוץ פשוט, ואז מקבל מול ערוץ פשוט. בשלב הבא אפשר להוסיף את מנגנון הרעש בערוץ, ולבסוף להוסיף את מנגנוני הקידוד והפיענוח. שימו לב: יש לתכנן מראש לפי פונקציונליות מלאה, ולהתחיל במימוש כאשר חלקים מסוימים ממומשים באופן טריוויאלי (placeholders). ניקוד חלקי יינתן על פונקציונליות חלקית לפי הסדר המתואר למעלה, אך לא על ביצוע שגוי של פונקציונליות מלאה.
- יש לבדוק ערך חזרה (return value) על כל קריאה למערכת (system call), ובמקרה של כישלון להדפיס הודעה אינפורמטיבית ל-`stderr`.

## בהצלחה!

## נספח: הוראות הגשה.

- התרגיל יוגש רק ע"י אחד השותפים.
- ההגשה דרך מודל בלבד.
- יש לשלוח קובץ ZIP המכיל את כל ה project של Visual C/C++. יש למחוק את ספריית DEBUG למעט exe.
- בדקו את ההגשה: שלחו אותה לעצמכם בדואר ובנו את האפליקציות על מחשב אחר (למשל במעבדה). נקודות יורדו על הגשה שלא נצליח לבנות בקלות.
- תנו שמות בעלי משמעות לקבצי ההרצה (למשל sender.exe).
- כתבו קוד קריא, כולל שמות בעלי משמעות למשתנים, פונקציות וכו'.
- יש לתעד את הקוד בקובץ README.txt או README.pdf. התייעוד יכיל את הפריטים הבאים:
  1. שמות שני המגישים ומספרי התלמיד שלהם.
  2. הוראות שימוש (כיצד להריץ), במידה ושונות מהתיאור למעלה.
  3. תיאור מבנה הקוד באופן כללי (מעין מפת דרכים למימוש), כולל הסבר איזה קוד לא נכתב על ידכם. (מותר ומומלץ להשתמש בספריות. **אסור** ליצור רושם כאילו כתבתם קוד שאחרים יצרו).
  4. באגים ידועים ומגבלות המימוש.
  5. אופציונלי: תיאור תוספות המזכות בבנוס, וגם תוספות אחרות אם יש.