

רשתות תקשורת – תשפ"ו – סמסטר א'

מטלה מספר 2 – מעבדת שכבת האפליקציה

מעבדת שכבת האפליקציה (Python 3.13+)

מטרת התרגיל

בתרגיל זה תלמדו את הבסיס לתכנות Socket ב-TCP בשפת Python ואת העקרונות של שכבת האפליקציה: כיצד לקוח ושרת מתקשרים בפרוטוקול פשוט שהוגדר על ידכם. אתם תיצרו socket, תקשרו אותו לכתובת ופורט מסוים ואז תשלחו ותקבלו פקטות. אין בשלב זה צורך לדעת כיצד כל אחד מפרוטוקולי שכבת התעבורה עובדים. מבחינתכם ברגע שה-socket נפתח, העברת המידע מתבצעת כקופסא שחורה.

בתרגיל זה תפתחו שני שרתים, במילים אחרות תהיו אחראים על שכבת האפליקציה שלהם:

- 1) שרת שמקבל ומטפל בבקשות של פרוטוקול חדש המטפל:
 - a. בביטויים אלגבריים. השרת צריך לקבל, לנתח ולפתור את הביטויים ולאחר מכן ליצור תגובה שמכילה את תוצאת הבקשה.
 - b. תומך בשליחת פרומפטים ל-GPT. השרת צריך לקבל פרומפט (סטרינג כלשהו, לא באמת משנה התוכן שלו) ולשלוח ל-GPT. לאחר מכן לשלוח חזרה תגובה המכילה את התשובה של ה-GPT. ניתן לממש את השרת באחת מהדרכים הבאות:
 - i. "דמו" של תשובה פשוטה/קבועה או הודעה מוסכמת של ה-GPT.
 - ii. לשלוח את הפרומפט ידני ל-GPT.
 - iii. לממש חיבור אמיתי ל-GPT דרך API.

ייתן בונוס על מימוש דינאמי הכולל חיבור אמיתי ל-CHAT.

- 2) שרת web פרוקסי שמאחסן את הבקשות ואת התגובות התואמות להם במטמון. אם בקשה נשלחה פעמיים בתוך טווח מסוים, השרת יגיש את התגובה המאוחסנת במטמון במקום לשלוח את הבקשה פעם נוספת.

מבנה המערכת

המערכת מורכבת ממספר רכיבים:

1. server.py - שרת TCP המקבל בקשות בפורמט JSON:

רשתות תקשורת – תשפ"ו – סמסטר א'

מטלה מספר 2 – מעבדת שכבת האפליקציה

- `mode="calc"`: חישוב ביטוי מתמטי אלגברי (באמצעות הפונקציה `safe_eval_expr`).
- `mode="gpt"`: שליחת פרומפט ל-GPT (קריאה אמיתית ל-OpenAI API או Stub/Fallback).
- כולל מנגנון מטמון (LRU) לתוצאות.
- 2. `client.py` - לקוח פשוט:
 - שולח בקשות לשרת ומדפיס תשובות.
 - מאפשר לבחור בין `mode calc --expr--` או `mode gpt --prompt--`.
- 3. `proxy.py` (רשות) – פרוקסי שקוף:
 - מעביר בקשות לשרת ומחזיר תשובות.
 - ניתן להרחיבו כך שיטפל במדיניות מטמון משלכם.
- 4. `protocol.md` - תיעוד הפורמט של הפרוטוקול JSON (בשורת טקסט אחת).
- 5. `/tests` - כולל טסט בסיסי.
- 6. `requirements.txt` - רשימת חבילות הדרושות:
 - `openai`
 - `python-dotenv`

משימות לביצוע

1. שרת חישובים (`server.py`):
 - השלם או שנה את מימוש הפונקציה `safe_eval_expr`.
 - הוסף טיפול בהמשכיות חיבור (תמיכה בבקשות מרובות על אותו חיבור): *השרת יכול להמשיך לקבל בקשות עד שהלקוח מתנתק. *חיבור חדש בכל פעם שהלקוח שולח בקשה לשרת אפשרי אך דורש סגירה מסודרת של חיבור ה-TCP. מימוש המשכיות יקבל ציון גבוה יותר.
2. לקוח (`client.py`):
 - אפשר ללקוח לבחור ביטוי מתוך רשימה מוגדרת מראש או להקליד ביטוי אלגברי/לוגי בעצמו ולשלוח בקשות נוספות לשרת/פרוקסי עד לרגע שבו יבחר לסיים את ההתקשרות.

רשתות תקשורת – תשפ"ו – סמסטר א'

מטלה מספר 2 – מעבדת שכבת האפליקציה

- תמוך במצב gpt - שימוש בפונקציית call_gpt.
- 3. פרוקסי (proxy.py):
 - השלם את הפונקציות proxy ו-client_handler.
 - הוסף טיפול בהמשכיות חיבור (תמיכה בבקשות מרובות על אותו חיבור): *השרת יכול להמשיך לקבל בקשות עד שהלקוח מתנתק. *חיבור חדש בכל פעם שהלקוח שולח בקשה לשרת אפשרי אך דורש סגירה מסודרת של חיבור ה-TCP. מימוש המשכיות יקבל ציון גבוה יותר.
 - הרחב את המימוש כך שהפרוקסי יוכל לשמור מטמון ולחסוך פניות מיותרות לשרת.
- 4. תמיכה ב-GPT:
 - ברירת מחדל של פונקציית Stub: המחזירה תשובת Fallback.
 - אפשרות: מימוש קריאה אמיתית ל-GPT בעזרת OpenAI API.
 - שמור את ה-API key בקובץ env. לפי התבנית הבאה:
OPENAI_API_KEY=your_api_key_here

פרוטוקול JSON

1. בקשת חישוב
לפי הדוגמה הבאה:

```
{"mode": "calc", "data": {"expr": "sin(2+3*4)"}, "options": {"cache": true}}
```
2. בקשת GPT
לפי הדוגמה הבאה:

```
{"mode": "gpt", "data": {"prompt": "כתוב תקציר קצר על רשת מחשבים"}, "options": {}}
```
3. תשובת שרת
לפי הדוגמה הבאה:

```
{"ok": true, "result": "...", "meta": {"from_cache": false, "took_ms": 7}}
```

הרצת הקוד

1. התקנת ספריות:
ב-bash:

```
pip install -r requirements.txt
```

רשתות תקשורת – תשפ"ו – סמסטר א'

מטלה מספר 2 – מעבדת שכבת האפליקציה

2. הפעלת שרת:

ב-bash:

```
python server.py --host 127.0.0.1 --port 5555 --cache-size 128
```

3. לקוח - שליחת בקשת חישוב:

ב-bash:

```
python client.py --host 127.0.0.1 --port 5555 --mode calc --expr "1+2*3"
```

4. לקוח - שליחת פרומפט ל-GPT:

ב-bash:

```
python client.py --host 127.0.0.1 --port 5555 --mode gpt --prompt "סכם את תורת ניוטון"
```

5. הרצת פרוקסי (אופציונלי):

ב-bash:

```
python proxy.py --listen-port 5554 --server-port 5555
```

```
python client.py --host 127.0.0.1 --port 5554 --mode calc --expr "sqrt(16)"
```

קבצי קוד שלד

מצורף עבורכם קבצי קוד שלד (skeletons) כדי שאתם תשתמשו בהם ותפתחו על בסיסם, אך זה לא חובה. דבר זה נועד על מנת למנוע בלבול, ולוודא שכולם עובדים מאותה נקודת פתיחה.

- server.py - שרת TCP עם JSON (כולל פונקציות `safe_eval_expr` ו-`call_gpt`, ה-`GPT stub/real call`).
- client.py - לקוח ששולח בקשות ומדפיס תגובות.
- proxy.py - שלד של פרוקסי (אתגר אופציונלי).
- tests/test_smoke.py - בדיקה קצרה שהמערכת עובדת.
- README.md - הוראות קצרות להרצה.
- requirements.txt - תלוי רק אם מפעילים GPT אמיתי.
- example.env - תבנית למפתח `OPENAI_API_KEY`.

הערות

- המערכת כולה מיועדת ל- Python 3.13+ עם התאמות לתיקון אזהרות `ast.Num` (שכבר אינו בשימוש).
- ניתן להרחיב את סט הפונקציות המתמטיות (פונקציית `safe_eval_expr`), לטפל בשגיאות נוספות או לשנות מדיניות מטמון.

רשתות תקשורת – תשפ"ו – סמסטר א'

מטלה מספר 2 – מעבדת שכבת האפליקציה

בנוסף יש לענות על השאלות הבאות:

1) נתב ה-Gateway הוא הנתב המחובר בין הרשת הפנימית לבין "שאר העולם" (האינטרנט). נתב ה-Gateway של הרשת הפנימית של הפקולטה למדעי הטבע מחובר לרשת הראשית של האוניברסיטה בקשר בקצב של X Gbps. הפקולטה הקצתה תקציב לשם קיצור הזמן שלוקח למשתמשים בפקולטה לגשת למידע של האוניברסיטה. התקציב מספיק לאחת משתי האפשרויות:

- לשפר את קצב החיבור לרשת האוניברסיטאית ל- $(X+2)$ Gbps.
- לקנות זיכרון המטמון, שיוצמד לנתב ה-Gateway. זמן הגישה של הנתב למטמון זניח.

מכיוון שהמרחקים הם קטנים, זמני ההתפשטות של האותות זניחים. נזניח גם את זמני העיבוד של הנתבים והשרתים.

מה צריך להיות ה-hit rate המינימלי של זיכרון המטמון (the fraction of requests that are satisfied by a cache), כדי שתהיה עדיפה האפשרות של קניית זיכרון מטמון?

2) ספק תוכן עולמי (כדוגמת טיקטוק, נטפליקס וכו') שוקל לפרוס בכל העולם. זיכרונות המטמון שיטפלו בכל הבקשות המגיעות מלקוחות, באחת משתי התצורות הבאות:

- Pull: אם הקובץ המבוקש נמצא במטמון, המטמון יבדוק אם הוא מעודכן, באמצעות Conditional Get. אם הקובץ המבוקש לא נמצא במטמון, המטמון יביא אותו מהשרת.
- Push: השרת סופר כמה פעמים מתקבלות בקשות לכל קובץ מסוים, ומדי פעם שולח לכל זיכרונות המטמון את הקבצים הפופולאריים ביותר (אותם קבצים נשלחים לכל זיכרונות המטמון). יתר על כן, בכל פעם שאחד הקבצים הפופולאריים מתעדכן, השרת שולח את הקובץ המעודכן לכל זיכרונות המטמון. לפיכך, אם קובץ נמצא במטמון – מובטח שזהו הקובץ המעודכן ביותר.

בחרו את ההיגד הנכון ונמקו:

- א. אם בכל העולם משתמשים נוטים לגשת לאותו קומץ הקבצים הפופולאריים, עדיף להשתמש בשיטת Push.
- ב. תמיד עדיף להשתמש בשיטת Pull.
- ג. תמיד עדיף להשתמש בשיטת Push.
- ד. אם הקבצים מאוד קטנים, כדאי להשתמש בשיטת Pull.

רשתות תקשורת – תשפ"ו – סמסטר א'

מטלה מספר 2 – מעבדת שכבת האפליקציה

מה להגיש

עליכם להגיש את כל קבצי הקוד המלאים של המערכת (כולל הוספת הערות ברורות בקוד). בנוסף לכך, יש להגיש מסמך PDF שיכלול את החלקים הבאים:

1. צילומי מסך עם הסברים
 - הראו את החלקים בקוד שהשלמתם או שיניתם.
 - צרפו הסברים קצרים שמבהירים מה עשיתם ולמה.
 2. תרשימי זרימה
 - תרשים זרימה המדגים את תהליך העבודה של מערכת המטמון כאשר הבקשה (התשובה לבקשה/התוצאה) לא קיימת במטמון.
 - תרשים זרימה נוסף המדגים את תהליך העבודה כאשר הבקשה (התשובה לבקשה/התוצאה) כן קיימת במטמון.
 - בכל תרשים יש להסביר את המעבר בין המצבים (לקוח → שרת/פרוקסי → מטמון → תשובה). הכוונה שאנחנו מבקשים תרשים שברור אילו חבילות עוברות בין מי ומה החלטה שכל אחד לוקח.
 3. צילומי מסך מתוך Wireshark (כולל הסברים לכל צילום):
 - 3.1. שרת בלבד:
 - הריצו את השרת.
 - הקליטו את הלקוח שולח את אותה בקשה פעמיים ברצף ולאחר מכן בקשה שונה.
 - 3.2. שרת + פרוקסי:
 - הריצו גם את השרת וגם את הפרוקסי.
 - הקליטו את הלקוח שולח את אותה בקשה פעמיים ברצף ולאחר מכן בקשה שונה (שימו לב: יש לשלוח לכתובת/פורט של הפרוקסי).
 - 3.3. שרת + פרוקסי + כיבוי שרת:
 - הריצו את השרת ואת הפרוקסי.
 - שלחו בקשה דרך הלקוח.
 - כבו את השרת.
 - שלחו שוב את אותה בקשה, ולאחר מכן בקשה שונה (גם כאן לכתובת/פורט של הפרוקסי).
- הערה: הקפידו להוסיף כמה צילומי מסך שצריך כדי להראות את ההתנהגות בבירור (אין ניכוי נקודות על "יותר מדי תמונות" כל עוד הן רלוונטיות). הקפידו גם לסנן את התוצאות כך שצילומי המסך שלכם יכילו רק פקטות/חבילות רלוונטיות.
- חשוב להראות שהפרוקסי אכן מחזיר תשובה מהמטמון, ולא פונה שוב לשרת.
4. תשובות לשאלות העיוניות

רשתות תקשורת – תשפ"ו – סמסטר א'

מטלה מספר 2 – מעבדת שכבת האפליקציה

- יש לענות על כל השאלות שניתנו במטלה, בצורה מסודרת וברורה.

5. הקלטות Wireshark

- יש לצרף את קבצי ההקלטה עצמם (.pcapng) יחד עם ההגשה.
- נדרשות לפחות שלוש הקלטות (בהתאמה לסעיפים 3.1–3.3).