

# 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 לתוכנות Socket ב-TCP בשפת Python ואת העקרונות של שכבת האפליקציה: כיצד לקוח ושרת מתקשורת בפרוטוקול פשוט שהוגדר על ידכם. אתם תיצרו socket, תקשרו אותו לכתובת ופורט מסוים ואז תשלחו ותקבלו פקודות. אין בשלב זה צורך לדעת כיצד כל אחד מפרוטוקולי שכבת התעבורה עובדים. מבחןתכם ברגע שהsocket נפתח, העברת המידע מתבצעת כקובץ שחורה.

בתרגיל זה תפתחו שני שירותים, במלילם אחריות היהו אחרים על שכבת האפליקציה שלהם:

- 1) שירות שמקבל ומטפל בבקשתו של פרוטוקול חדש המטפל:
  - a. בביטויים אלגבריים. השירות צריך לקבל, לנתח ולפטור את הביטויים ולאחר מכן ליצור תגובה שמכילה את תוצאת הבקשה.
  - b. תומך בשליחת פרומפטים ל-GPT. השירות צריך לקבל פרומפט (סטרינג כלשהו, לא באמת משנה התוכן שלו) ושלוח ל-GPT. לאחר מכן לשולח חזרה תגובה המכילה את התשובה של ה-GPT.
- ניתן למש את השירות באמצעות מהדרכים הבאים:
  - i. "דמו" של תשובה פשוטה/קבועה או הודעה מוסכמת של ה-GPT.
  - ii. לשולח את הפרומפט ידני ל-GPT.
  - iii. למש חיבור אמיתי ל-GPT דרך API.

### יינתן בונוס על מימוש דינאמי הכלל חיבור אמיתי ל-CHAT.

- 2) שירות web פרוקטי שמאחסן את הבקשות ואת התשובות התואמות להם במטמון. אם בקשה נשלחה פעמיים בתור טווח מסוים, השירות יגיש את התגובה המאוחסנת במטמון במקום לשולח את הבקשה פעם נוספת.

### מבנה המערכת

המערכת מורכבת ממספר רכיבים:

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

# רשתות תקשורת – תשפ"ו – סמסטר א'

## מטלה מספר 2 – מעבדת שכבת האפליקציה

- mode: חישוב ביטוי מתמטי אלגברי (באמצעות הפונקציה `safe_eval_expr`).
  - OpenAI: שליחת פרומפט ל-GPT (קריאה אמיתית ל-OpenAI API-Stub/Fallback או `mode="gpt"`).
  - כולל מנגנון מטמון (LRU) לתוצאות.
- client.py - ל��וח פשוט:
  - שולח בקשות לשרת ומדפיס תשובות.
  - מאפשר לבחור בין `mode calc` – `mode gpt` – `--prompt` – `--expr` – `--eval` – `--proxy`.
- proxy.py (רשות) – פרוקסי שկוף:
  - מעביר בקשות לשרת ומחזיר תשובות.
  - ניתן להרחיבו כך שיטיפול במידיניות מטמון משלכם.
- protocol.md - תיעוד הפורמט של הпрוטוקול JSON (בשורות טקסט אחד).
  - tests/ – כולל טסט בסיסי.
- requirements.txt - רשימה של חבילות הדרישות:

openai ◦  
python-dotenv ◦

### משימות לביצוע

1. שרת חישובים (server.py):
  - השלם או שנה את מימוש הפונקציה `safe_eval_expr`.
  - הוסף טיפול בהMSCיות חיבור (תמייה בבקשתות רבות על אותו חיבור)\*: השרת יכול להמשיך לקבל בבקשתות עד שהלקוח מתנתק.  
\*חיבור חדש בכל פעם שהלקוח שולח בקשה לשרת אפשרי אך דרוש סגירה מסודרת של חיבור ה-TCP. מימוש המשיכות יקבל ציון גובה יותר.
2. ל��וח (client.py):
  - אפשר ללקוח לבחור ביטוי מתוך רשימה מוגדרת מראש או להקליד ביטוי אלגברי/לוגי בעצמו ולשלוח בבקשתות נוספות לשרת/פרוקסי עד לרגע שבו יבחר לסיים את ההתקשרות.

# רשתות תקשורת – תשפ"ו – סמסטר א'

## מטלה מספר 2 – מעבדת שכבת האפליקציה

- תמור במצב `gpt` - שימוש בפונקציית `gpt_call`.
  - פרוקסי (proxy.py):
    - השלם את הפונקציות `proxy` ו-`client_handler`.
    - הוסף טיפול בהמשכיות חיבור (תמייה בבקשתים רבות על אותו חיבור): השרת יכול להמשיך לקבל בבקשת עד שהלקוח מתנתק.
    - \*חיבור חדש בכל פעם שהלקוח שולח בקשה לשרת אפשרי אף דרוש סגירה מסודרת של חיבור ה-TCP. שימוש המשכיות יקבל ציון גבואה יותר.
  - הרחב את השימוש כך שהפרוקסי יוכל לשמור מטמון ולהסור פניות מיותרות לשרת.
4. תמייה ב-GPT:
- ברירת מחדל של פונקציית `Stub`: המחזירה תשובה `Fallback`.
  - אפשרות: שימוש קריאה אמיתית ל-GPT באמצעות API OpenAI.
  - שומר את ה-`key` API בקובץ `.env` לפי התבנית הבאה:

`OPENAI_API_KEY=your_api_key_here`

### פרוטוקול JSON

#### 1. בקשה חישוב

לפי הדוגמה הבאה:

```
{"mode": "calc", "data": {"expr": "sin(2+3*4)"}, "options": {"cache": true}}
```

#### 2. בקשה GPT

לפי הדוגמה הבאה:

```
{"mode": "gpt", "data": {"prompt": "כתוב תקציר קצר על רשות מחשבים"}, "options": {}}
```

#### 3. תשובה שרת

לפי הדוגמה הבאה:

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

### הרכבת 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:

:bash

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

3. ל��ח - שליחת בקשת חישוב:

:bash

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

4. ל��ח - שליחת פרוםפט ל-GPT

:bash

```
"סכם את תורה ניוטון"
```

5. הרצת פרוקסי (אופציונלי):

: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, ולודא שגםם עובדים מאותה נקודת פתיחה.

- safe\_eval\_expr.py - שרת TCP עם JSON (כולל פונקציות `eval`, `safe_eval`, `call_gpt`, `call_real`).
- client.py - ל��ח ששלוח בקשות ומדפיס תשובות.
- 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.
- OPENAI\_API\_KEY.example.env - תבנית למפתח KEY.

### הערות

- המערכת יכולה מיעדרת ל- Python 3.13+. עם התאמות לתיקון זההroat `ast.Num` (שכבר אין בשימוש).
- ניתן להרחיב את סט הפונקציות המתמטיות (פונקציית `eval`)safe\_eval\_expr, לטפל בשגיאות נוספות או לשנות מדיניות מתמון.

# רשתות תקשורת – תשפ"ו – סמסטר א'

## מטלה מספר 2 – מעבדת שכבת האפליקציה

בנוסף יש לענות על השאלות הבאות:

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

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

מכיוון שהמרחקים הם קטנים, زمنי הרתפשות של האותות זניחים. נניחס גם את زمنי העבודה של הנתבים והשרטים.

מה צריך להיות hit rate המינימלי של זיכרון המטמון (the fraction of requests that are satisfied by a cache memory)?

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

- Pull: אם הקובץ המבוקש נמצא במטמון, המטמון יבודק 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, ומיד פעם שולח לכל זיכרונות המטמון את הקבצים הפופולריים ביותר (אותם קבצים נשלחים לכל זיכרונות המטמון). יתר על כן, בכל פעם שאחד הקבצים הפופולריים מתעדכן, השרת שולח את הקובץ המעודכן לכל זיכרונות המטמון. לפיכך, אם קובץ נמצא במטמון – מובטח שהוא הקובץ המעודכן ביותר.

- בחרו את הrigid הנכון ונמקו:
- א. אם בכל העולם משתמשים נוטים לגשת לאותו קומץ הקבצים הפופולריים, עדיף להשתמש בשיטת Push.
  - ב. תמיד עדיף להשתמש בשיטת Pull.
  - ג. תמיד עדיף להשתמש בשיטת Push.
  - ד. אם הקבצים מאד 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 העروת 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 שمبיארים מה עשיתם ולמה.
2. תרשימי זרימה
  - תרשימים זרימה המדגים את תהליך העבודה של מערכת המטמון כאשר הבקשה (התשובה לבקשת/התוצאה) לא קיימת במטמון.
  - תרשימים זרימה נוספת המדגים את תהליך העבודה כאשר הבקשה (התשובה לבקשת/התוצאה) כן קיימת במטמון.
  - בכל תרשימים יש להסביר את המעבר בין המצביעים (לקוח → שרת/פרוקסי → מטמון → תשובה). הכוונה שאנוחנו מבקשים תרשימים שבורור אילו חבילות עוברות בין מי ומה החלטהiscal אחד לוקח.
3. צילומי מסך מתוך Wireshark (כולל הסברים לכל צילום):

#### 3.1. שרת בלבד:

- הרכזו את השרת.
- הקליטו את הלkop שולח אותה בקשה פעמיים בראף ולאחר מכן בקשה שונה.

#### 3.2. שרת + פרוקסי:

- הרכזו גם את השרת וגם את הפרוקסי.
- הקליטו את הלkop שולח אותה בקשה פעמיים בראף ולאחר מכן בקשה שונה (שים לב: יש לשולח לכתובת/פורט של הפרוקסי).

#### 3.3. שרת + פרוקסי + כיבוי שרת:

- הרכזו את השרת ואת הפרוקסי.
- שלחו בקשה דרך הלkop.
- כבו את השרת.
- שלחו שוב את אותה בקשה, ולאחר מכן בקשה שונה (גם כאן לכתובת/פורט של הפרוקסי).

הערה: הקפידו להוסיף כמה צילומי מסך שציריך כדי להראות את ההתנהגות בבחירה (אין ניכוי וקודות על "יותר מדי תמנונות" כל עוד הן רלוונטיות). הקפידו גם לסנן את התוצאות כך שצלומי המסך שלכם ייכלו רק פקטות/חbillות רלוונטיות.

חשוב להראות שהפרוקסי אכן מוחזר תשובה מהמטמון, ולא פונה שוב לשרת.

#### 4. תשובות לשאלות העיוניות

## **רשות 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 לצלף את קבצי הרקלה עצם (.pcapng) יחד עם ההגשה.
  - נדרשות לפחות שלוש הקלטות (בהתאם לסעיפים 3.3–3.1).