

רשתות תקשורת – תשפ"ג – סמסטר א' – מטלה 2

את מטלה זו יש להגיש בזוגות כקובץ zip עם מספרי ת"ז של הסטודנטים/ות. לתיבת ההגשה במודל. הגשות באיחור יתאפשרו עד 4 ימים כאשר לכל יום איחור ירדו 5 נקודות. שימו לב, יש להגיש קבצי קוד, pcaps וגם קובץ pdf המכיל צילומי מסך (בכל מקום שאתם עושים משהו תצלמו מסך ותסבירו איך הגעתם למסקנה).

- (1) את המטלה יש להגיש עד התאריך המצוין בתיבת הגשה.
- (2) יש להגיש את כל קבצי המטלה (קוד, פלט תעבורה, הסבר) כולל הסברים שלכם והקלטות Wireshark דחוסים לקובץ zip ששמו הוא מס' ת.ז. של המגישים עם קו תחתון בניהם ID_ID.
- (3) הגשה בזוגות הינה אפשרית אך לא חובה.
- (4) מותר לכם להשתמש בכל החומר שנמצא במודל כולל קוד בתרגילים. חומרים אחרים אין אפשרות. כמובן שאפשר להיעזר באינטרנט להבנה של תהליכים וקוד אבל בשום פנים ואופן לא להעתיק קוד.
- (5) אין איחורים ללא אישור מיוחד, איחור ללא אישור יגרור אפס אוטומטי.
- (6) הגשת העבודות תתבצע דרך מערכת ה Moodle של הקורס (לא דרך האימייל).
- (7) יש להקפיד על כללי עיצוב הקוד שנלמדו בתואר (נא להקפיד על פלט ברור, הערות קוד במידה ושמות משתנים בעלי משמעות). קוד רץ בלבד יכול לקבל לכל היותר ציון 60, שאר 40 הנקודות זה הסברים שלכם, ידע, קוד קריא וכו'.
- (8) ניתן להגיש תרגילים למערכת מספר בלתי מוגבל של פעמים כאשר כל הגשה דורסת את הקודמת.
- (9) העבודה הינה אישית של הזוג ואסור לקבל עזרה מאנשים מחוץ לאוניברסיטה או בתוכה לה. אנשים המתקשים ורוצים עזרה יכולים לפנות לצוות הקורס בשעות הקבלה או להעלות שאלה לאתר הקורס.
- (10) אסור להעביר קטעי קוד בין סטודנטים, להעלות פתרונות או חלקי פתרונות לאתרים ברשת האינטרנט, פורומים או בקבוצות תקשורת שונות.
- (11) סטודנטים שיעתיקו פתרון, יקבלו 0 בכל המטלות בקורס ונעלה דיווח לוועדת המשמעת המוסדית.

נכתב בלשון זכר מטעמי נוחות בלבד, אך מתייחס לנשים ולגברים כאחד.

מעבדת שכבת האפליקציה (Python 3.8+)

בתרגיל זה, תלמדו את הבסיס לתכנות socket של Python. 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 אז העברת המידע הינו קופסא שחורה.

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

- 1) שרת שמקבל ומטפל בבקשות של פרוטוקול חדש המטפל בביטויים אלגבריים. השרת צריך לקבל לנתח ולפתור את הביטויים ולאחר מכן ליצור תגובה שמכילה את תוצאת הבקשה.
- 2) שרת וואב (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.

הקוד

מצורף עבורכם 5 קבצים. בשלושה מהקבצים (server.py, client.py, proxy.py) צריך לשנות ולערוך את הקוד בהתאם. שני הקבצים הנותרים (calculator.py ו-api.py) הם קבצים שמגדירים את המערכת ואין שום צורך לשנות בהם דבר.

הלקוח (client.py)

בקובץ הזה רשמנו לקוח פשוט אשר בונה ביטויים המוגדרים מראש, שולח את הבקשה אל השרת הפרוקסי, ואז מעבד את התגובה. לפי חלק מן השאלות אתם תצרכו לשנות חלק מהערכים של המשתנים שממוקמים בצורה נוחה ב-main.

השרת (server.py)

בקובץ הזה כתבנו את קוד השלד של השרת. אתם תצטרכו להשלים את הקוד. החלקים שצריך להשלים קוד מסומנים ב #Fill in start ו #Fill in end (הערה: בחלק מהמקומות צריך להשלים את השורה ובחלק צריך להשלים שורה אחת או יותר).

השרת מקבל ומעבד (מחשב) את הבקשה ולאחר מכן שולח חזרה את התגובה לפי הבקשה. הקוד שמעבד את הבקשה נכתב כבר ולכן הקוד שאתם צריכים להשלים מסומן בפונקציות הבאות: "server" ו "client_handler". אתם יכולים לשנות את שני המשתנים שנמצאים בראש הקובץ בשביל לשנות איך המטמון יטפל בתגובות של השרת (לא נדרש על ידי השאלות).

הפרוקסי (proxy.py)

בקובץ הזה כתבנו את קוד השלד של השרת. אתם תצטרכו להשלים את הקוד. החלקים שצריך להשלים קוד מסומנים ב #Fill in start ו #Fill in end (הערה: בחלק מהמקומות צריך להשלים את השורה ובחלק צריך להשלים שורה אחת או יותר).

הפרוקסי מקבל בקשה, בודק אם הוא נמצא כבר במטמון, אם לא, הפרוקסי מעביר את הבקשה לשרת ואז מאחסן במטמון את התגובה לפי הבקשה המקורית ולפי התגובה של השרת (ניתן לשלוט על התנהגויות יותר ספציפיות של המטמון על ידי שינוי

ערכים בראש הפרוטוקול). הקוד שמטפל במטמון נרשם כבר ולכן הקוד שאתם צריכים להשלים מסומן בפונקציות הבאות:
 "proxy" ו "client_handler".

הפרוטוקול

להלן כותרת המקטע של הפרוטוקול והסבר על כל אחד מהשדות שלו, באמצעות זה נוכל לשלוח, לקבל ולפענח את החבילה.

0								1								2								3								בית
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	סיבית
זמן יוניקס																																
אורך כולל																שמור 0 0 0				C	S	T	קוד מצב									
שליטת\שליפת מטמון																ריפוד																
המידע																																

זמן יוניקס (32 סיביות)

הזמן שהפקטה נוצרה בשניות מאז 1970-01-01 00:00:00 UTC.

אורך כולל (16 סיביות)

האורך הכולל של הפקטה בבתים, כולל גם הפתיה וגם המידע. גודל הפקטה המינימאלי הוא 12 בתים (פקטה עם רק פתיה וללא מידע). נגדיר את גודל הפקטה המקסימאלי להיות 8192 בתים (כלומר אורך המידע יכול להיות לכל היותר 8180 בתים).

שמור (3 סיביות)

שדה שמור לשימוש עתידי ואמור להיות אפס.

דגלים (3 סיביות)

- Cache (1 סיבית): מסמל שאנחנו רוצים לשמור את הפקטה במטמון, או שהפקטה נשמרה במטמון. (0 = אל תשמור/לא נשמר במטמון, 1 = תשמור/נשמר במטמון)
- Steps (1 סיבית): מסמל שאנחנו רוצים לכלול את צעדי החישוב בתשובה, או שהתשובה מכילה את צעדי החישוב. (0 = אל תכלול/לא כולל את צעדי החישוב, 1 = תכלול/כלול את צעדי החישוב)
- Type (1 סיבית): מסמל שהפקטה היא בקשה. (0 = תגובה, 1 = בקשה)

קוד מצב (10 סיביות)

קוד מצב של התגובה

2xx = Success

4xx = Client Error

5xx = Server Error

0 = not a response.

שליטת\שליפת מטמון (16 סיביות)

(הערה: אם הדגל C הוא 0 אז מתעלמים מהערך של השדה הזה).

(הערה: אם הערך של השדה הוא הערך המקסימאלי של unsigned-integer בעל 16 סיביות [65535], נפרש את זה כשמירה למטמון ללא הגבלת זמן).

1. לתגובות ערך השדה הוא הזמן המקסימאלי שהתגובה יכולה להיות במטמון (בשניות).

- אם ערך השדה הוא 0, התגובה לא תשמר במטמון.

2. לבקשות ערך השדה הוא הזמן המקסימאלי שתגובה יכולה להיות במטמון שהלקוח מוכן לקבל (בשניות). כלומר

המטמון לא יחזיר תגובות השמורות במטמון יותר מערך השדה, במצב זה המטמון פשוט ישלח את הבקשה מחדש.

- אם ערך השדה הוא 0, אנחנו מכריחים את המטמון לטעון מחדש את המטמון עבור הבקשה הזאת.

ריפוד (16 סיביות)

ריפוד בשביל לוודא שהפתיח נגמר בגבול של 32 סיביות והמידע מתחיל ובגבול 32 סיביות.

מידע (לכל היותר 8180 בתים = 65440 סיביות)

המידע של ההודעה.

הרצת הקוד

כאשר מריצים את הלקוח, פרוקסי והשרת ניתן להשתמש בדגל "-h" (או "--help") בשביל לראות איזה הגדרות אפשר לשנות (לדוגמה, python <file's name>.py -h).

הלקוח כברירת מחדל שולח את הבקשה לפורט של השרת, לכן כאשר אתם רוצים לשלוח בקשה לפרוקסי צריך להוסיף את הארגומנט "<proxy's port> -p" (לדוגמה, "python client.py -p <proxy's port>"). כפי שנרשם לעיל, בתוכנית הראשית אפשר לשנות את הביטוי ששולחים באמצעות ההערות (לשים בהערה את הביטוי הנוכחי ולבטל הערה של ביטוי אחר), ובנוסף לכך ניתן גם לשנות את הערך של המשתנים על מנת לשנות את הערך של השדה התואם בפתיח של הבקשה.

בשרת גם כן (נמצא בראש הקובץ) ניתן לשנות את ערך המשתנים בשביל לשנות את ערך השדות התואמים בפתיח של התגובה.

הערה: הקובץ "api.py" מכיל (נמצא בראש הקובץ) כמה משתנים שמגדירים את ערך ברירת המחדל של כתובת ופורט גם של השרת וגם של הפרוקסי. אם בפירוש לא רשמתם בארגומנטים של השרת ו\או של הפרוקסי הערכים האלו הם ברירת המחדל. (פורט ברירת המחדל של השרת הוא 9999 ופורט ברירת המחדל של הפרוקסי הוא 9998).

מה להגיש

אתם תגישו את הקוד המלא של server.py ושל proxy.py (תוודאו להוסיף הערות לחלקים שהשלמתם שמסבירים מה עשיתם). בנוסף לכך תגישו מסמך pdf שמכיל את הדברים הבאים:

1. צילומי מסך של החלקים שמילאתם עם הסברים.
 2. תרשים זרימה הממחיש את מערכת המטמון עם הסברים.
 3. צילומי מסך של הפקטות **הרלוונטיות** מהקלטות ה Wireshark הבאות (אל תשכחו להוסיף הסברים):
 - 3.1. הריצו את השרת. תקליטו את הלקוח שולח את אותה הבקשה פעמיים אחד אחרי השני ולאחר מכן שולח בקשה שונה.
 - 3.2. הריצו את השרת ואת הפרוקסי. תקליטו את הלקוח שולח את אותה הבקשה פעמיים אחד אחרי השני ולאחר מכן שולח בקשה שונה. (זכרו לשלוח את זה לפרוקסי על ידי שינוי הפורט אצל הלקוח).
 - 3.3. הריצו את השרת ואת הפרוקסי. תקליטו את הלקוח שולח בקשה, לאחר מכן סיגרו את השרת, תשלחו את אותה הבקשה עוד הפעם ולאחר מכן שילחו בקשה שונה. (זכרו לשלוח את זה לפרוקסי על ידי שינוי הפורט אצל הלקוח).
- (הערה: כאשר מוסיפים את צילומי המסך של הפקטות, אל תפחדו להוסיף כמה צילומי מסך של הקלטה אחת, אנחנו לא נוריד נקודות על הוספת יותר מידי תמונות **רלוונטיות**, אבל כן נוריד אם לא מוסיפים מספיק. תוודאו שצילומי המסך מראים שהפרוקסי באמת שמר את הבקשה למטמון ולא שלח שוב את הבקשה לשרת).
- אל תישכחו להגיש גם את הקלטות ה Wireshark! (אמורות להיות **לפחות** 3 הקלטות).