

Homework 3 Dry

Due Date: 22.6.2023

Teaching assistant in charge: Harel

Important: the Q&A for the exercise will take place at a public forum Piazza only. Critical updates about the HW will be published in pinned notes in the piazza forum. These notes are mandatory and it is your responsibility to be updated. A number of guidelines to use the forum:

- Read previous Q&A carefully before asking the question; repeated questions will probably go without answers
- Be polite, remember that course staff does this as a service for the students
- You're not allowed to post any kind of solution and/or source code in the forum as a hint for other students; In case you feel that you have to discuss such a matter, please come to the reception hour
- When posting questions regarding hw3 , put them in the hw3 folder

Only the TA in charge can authorize postponements. In case you need a postponement, fill in this form - <https://forms.office.com/r/cCA9WHhVqw>

Dry part submission instructions:

1. Please submit the dry part to the electronic submission of the dry part on the course website.
2. The dry part submission must contain a single dry.pdf file containing the following:
 - a. The first page should contain the details about the submitters - Name, ID number and email address.
 - b. Your answers to the dry part questions.
3. Only typed submissions will be accepted. Scanned handwritten submissions will not be accepted.
4. Only PDF format will be accepted.
5. You do not need to submit anything in the course cell.
6. When you submit, **retain your confirmation code and a copy of the PDF**, in case of technical failure. It is **the only valid proof** of your submission.

יש לנמק כל תשובה אלא אם במפורש נאמר אחרת, תשובות ללא נימוק לא יתקבלו.

שאלה 2 - Networking - תקשורת (52 נק')

חלק פתוח: הסבירו בקצרה (לא יותר מ-2 שורות!)
לאחר מסע בחופי ישראל, יוליה כלבת הים הנזירית שמעה על מסיבת הקיץ של הפקולטה למדעי המחשב, ה-"טאוביץ", והחליטה לקפוץ לביקור. כשהגיעה לבניין טאוב, הבינה יוליה כי לא תמצא פה חוף ים וחיפשה דרכים לצנן את גופה. יוליה החליטה להזמין את קראנץ' הפיסטוק שכולם מדברים עליו וניסתה להזמין לעצמה אחד, אך אבוי, יוליה היא כלבת ים ולא עברה עדיין את הקורס מערכות הפעלה ולכן לא יודעת כיצד עובדת רשת האינטרנט, עזרו ליוליה להבין מושגים בסיסיים בעולם הרשתות על מנת שתוכל להזמין לעצמה קראנץ' פיסטוק.

א. (4 נקודות) הסבירו מה תפקיד של פרוטוקול ARP.

ב. (4 נקודות) איזה מידע הלקוח צריך לדעת על השרת לפני ההתחברות?

ג. (4 נקודות) איזה מידע הלקוח ידע על השרת אחרי ההתחברות?

ד. (4 נקודות) איזה מידע השרת צריך לדעת על הלקוח לפני ההתחברות?

ה. (4 נקודות) איזה מידע השרת ידע על הלקוח אחרי ההתחברות?

ו. (6 נקודות) מה הבדל בין הפורט (port) שבשימוש השרת וזה של הלקוח. אין נבחר כל אחד מהם?

ז. (6 נקודות) מה הבדל בין פרוטוקול TCP ו-UDP? הסבירו למה חלק מהאפליקציות מעדיפות TCP וחלק UDP.

ח. (6 נקודות) מהו תפקיד פרוטוקול ה-DNS?

- א. לשלוח פקטות (frame) מחשבי קצה בתוך אותה רשת (LAN connectivity).
- ב. לתרגם כתובת IP לכתובת MAC.
- ג. לתרגם שם השרת לכתובת IP.
- ד. לתרגם שם השרת לכתובת MAC.
- ה. לשלוח פקטות בין מחשבי קצה ברשתות שונות (WAN).
- ו. לאפשר תקשורת בין שני תהליכים במחשבי קצה ברשתות שונות (WAN).

נימוק:

ט. (8 נקודות) מהו תפקיד פרוטוקול ה-NAT?

- א. וידוי של הצפנת המידע.
- ב. שימוש של מספר קטן של כתובות IP עבור הרבה מכשירים בתוך הרשת.
- ג. הסתרת זהות הלקוח.
- ד. הסתרת זהות השרת.
- ה. וידוי של הצפנת המידע + שימוש של מספר קטן של כתובות IP עבור הרבה מכשירים בתוך הרשת.
- ו. וידוי של הצפנת המידע + הסתרת זהות הלקוח.

נימוק:

י. (6 נקודות) מה נכון במודל תקשורת שרת/לקוח על מנת ליצור connection (חיבור)?

- א. הלקוח חייב לדעת גם שם של ה-domain של השרת וגם מספר הפורט של השרת.
- ב. שרת חייב לדעת כתובת IP של הלקוח, אך הלקוח לא חייב לדעת כתובת IP של השרת.
- ג. שרת חייב לדעת כתובת IP של הלקוח, וגם הלקוח חייב לדעת כתובת IP של השרת.
- ד. השרת חייב לדעת גם כתובת IP וגם מספר הפורט של הלקוח.
- ה. הלקוח חייב לדעת כתובת שם של ה-domain של השרת. הפורט הינו קבוע לפי סוג ה-application.
- ו. המידע הנחוץ תלוי בצד שיוזם את החיבור.

שאלה 2 - סינכרון (48 נק')

לאחר הפרידה המתקשרת של נוגה (מוכרת בעיקר על ידי השיר שלה, "חד קרנל") ומרבי, עולם הפופ הישראלי התחלק לשתי קבוצות, קבוצת נוגה וקבוצת מרבי. בין הקבוצות שררה שנאה רבה ולא היו מוכנים לשהות באותו החדר, ולכן הוגדר כי כאשר חבר אחת הקבוצות רוצה להיכנס לחדר מסויים עליו לציית לכלל הבא: אם יש חברי קבוצה אחרת בחדר אזי אסור לו להיכנס ועליו להמתין עד שיעזבו (לעומת זאת, מספר חברים מאותה הקבוצה יכולים לשהות בחדר באותו הזמן).

סמני נכון / לא נכון (אין צורך להסביר):

1. (3 נק') יכולים להיות שני חברים מקבוצות שונות באותו חדר במקביל: נכון / לא נכון
2. (3 נק') יכולים להיות שני חברים מאותה הקבוצה בחדר במקביל: נכון / לא נכון
3. (3 נק') חברי קבוצה אחת עלולים להרעיב (כניסת) חברי קבוצה אחרת: נכון / לא נכון

בסעיפים הבאים מוצג קוד למימוש כניסה ויציאה של חברים בקבוצות השונות אל ומחדר מסוים, כאשר נתון כי:

- כל חוט מייצג חבר קבוצה כלשהי.
- בכניסה לחדר חבר הקבוצה קורא ל `onArrival(int team)`, שמקבלת את הקבוצה אליה שייך.
- ביציאה מהחדר חבר הקבוצה קורא ל `onLeave(int team)` שמקבלת את הקבוצה אליה שייך.
- הערכים 0 ו-1 של `team` מייצגים את קבוצת מרבי וקבוצת נוגה, בהתאמה.
- (הניחו שאמצעי הסנכרון עברו אתחול תקין והתעלמו מבעיות קומפילציה אם ישנן, שכן מטרת השאלה אינה לבדוק שגיאות אתחול/תחביר).

1. <code>#include <pthread.h></code>	11. <code>void onArrival(int team) {</code>
2. <code>int members = 0;</code>	12. <code>mutex_lock(&global);</code>
3. <code>mutex_t global;</code>	13. <code>while (members > 0) {</code>
4. <code>void onLeave(int team) {</code>	14. <code>mutex_unlock(&global);</code>
5. <code>mutex_lock(&global);</code>	15. <code>sleep(10);</code>
6. <code>members --;</code>	16. <code>mutex_lock(&global);</code>
7. <code>mutex_unlock(&global);</code>	17. <code>}</code>
8. <code>}</code>	18. <code>mutex_unlock(&global);</code>
9. <code>}</code>	19. <code>members++;</code>
10. <code>}</code>	20. <code>}</code>

1. (12 נק') בהתייחס לקוד הנ"ל, הקיפי את כל התשובות הנכונות (עשויה להיות יותר מאחת). עבור כל תשובה שהקפת, תארי דוגמת הרצה המובילה לתשובה זו.

- a. קיימת בעיית נכונות עקב race condition למשאבים משותפים.
- b. קיימת בעיית DeadLock / Livelock בקוד.
- c. הקוד משתמש ב-Busy Wait שפוגע בנצילות המעבד.
- d. הקוד מפר את כלל הכניסה לחדר (שהוגדר בתחילת השאלה).

נימוק:

המימוש של כניסה ויציאה שונה כך שישתמש במשתני תנאי:

```
1  int members[2] = {0}; // 2 counters
2  cond_t conds[2];      // 2 condition variables
3  mutex_t global;
4  void onArrival(int team) {
5      mutex_lock(&global);
6      int other = team? 0 : 1;
7      while(members[other] > 0)
8          cond_wait(&conds[team] , &global);
9      members [team]++;
10     mutex_unlock(&global);
11 }
12 void onLeave(int team) {
13     mutex_lock(&global);
14     members [team]--;
15     int other = team? 0 : 1;
16     cond_signal(&conds[other]);
17     mutex_unlock(&global);
18 }
```

- אך עומר (עתודאי במדמ"ח) טען שקוד זה גורם לחוסים להתעורר שלא לצורך ומיד לחזור למצב המתנה.
1. (7 נק') הסבירי את טענתו של עומר באמצעות דוגמת ריצה קונקרטית.

1. (8 נק') כיצד ניתן לתקן את הבעיה שהציג עומר בסעיף הקודם?

עומר ניסה לשפר עוד את יעילות הקוד והחליט להשתמש בשני מנעולים: מנעול ראשון בעבור חברי קבוצה הנכנסים לחדר, ומנעול שני בעבור חברי קבוצה היוצאים מהחדר. להלן המימוש החדש (השינויים בקוד מודגשים):

```
1 int members [2] = {0};           // 2 counters
2 cond_t conds[2];                 // 2 condition variables
3 mutex_t m_arrival, m_leave;     // there are *2* locks now
4 void onArrival(int team){
5     mutex_lock(&m_arrival);
6     int other = team? 0 : 1;
7     while(members [other] > 0)
8         cond_wait(&conds[faculty] , &m_arrival);
9     int tmp = members [team];
10    members [team] = tmp + 1;
11    mutex_unlock(&m_arrival);
12 }
13 void onLeave(int team){
14     mutex_lock(&m_leave);
15     int tmp = students[team];
16     members [team] = tmp - 1;
17     int other = team? 0 : 1;
18     cond_signal(&conds[other]);
19     mutex_unlock(&m_leave);
20 }
```

1. (12 נק') בהתייחס לקוד הנ"ל, הקיפי את כל התשובות הנכונות (עשויה להיות יותר מאחת).
עבור כל תשובה שהקפת, תארי דוגמת הרצה המובילה לתשובה זו.

- a. יתכנו 2 חברים מקבוצות שונות בתוך החדר ביחד, עקב race condition משותף.
- b. יתכן מצב שחבר קבוצה כלשהי לא נכנס לחדר למרות כלל הכניסה שמתיר זאת, עקב race condition למשאב משותף.
- c. קיימת בעיית DeadLock / Livelock בקוד.
- d. יתכן מצב שחבר קבוצה כלשהי יחכה למרות כלל הכניסה שמתיר זאת, כאשר אין race condition למשאב משותף.

נימוק:

