

ראני וישאחי: 308198134

עאמר אווס: 322998923

89/100

שאלה 1 - Networking - תקשורת (52 נק')

- א. (4 נקודות) הסבירו מה תפקיד של פרוטוקול ARP. **התפקיד של הפרוטוקול הוא למצוא ה-MAC ADDRESS של מכונה אחרת בהינתן ה-IP שלה בתוך רשת לוקאלית.**
- ב. (4 נקודות) איזה מידע הלקוח צריך לדעת על השרת לפני ההתחברות? **כתובת ה-IP וה-PORT שהשרת מאזין עליהן כפי שראינו בהרצאה שני הנתונים משמשים לקבוע את המכונה(שכבה 3) ואת האפליקציה שרוצים לגשת אליה.**
- ג. (4 נקודות) איזה מידע הלקוח ידע על השרת אחרי ההתחברות? **כלום, הלקוח יש לו את כל המידע שהוא צריך.**
- ד. (4 נקודות) איזה מידע השרת צריך לדעת על הלקוח לפני ההתחברות? **כלום, הלקוח הוא זה ששולח את הבקשה להתחברות לשרת יחד עם המידע הרלוונטי.**
- ה. (4 נקודות) איזה מידע השרת ידע על הלקוח אחרי ההתחברות? **כתובת ה-IP וה-PORT של הלקוח, מידע זה הכרחי כדי לקבוע את המכונה והאפליקציה שצריכים לקבץ את המידע שיישלח.**
- ו. (6 נקודות) מה הבדל בין הפורט (port) שבשימוש השרת וזה של הלקוח. אין נבחר כל אחד מהם? **ה-PORT של השרת הוא ערך קבוע ואינו משתנה, הלקוח מקבל מספר PORT חדש בכל פעם שהוא מתחבר מחדש. איך נבחר כל אחד מהם?**
2
- ז. (6 נקודות) מה הבדל בין פרוטוקול TCP ו-UDP? הסבירו למה חלק מהאפליקציות מעדיפות TCP וחלק UDP. **TCP הוא reliable כלומר שהוא מבטיח הגעת כל המידע שנשלח (כל packet) ובסדר השידור לעומת UDP שמאפשר איבוד מיעד או הגעה בסדר שונה מזה של השליחה. מועדף להשתמש ב-TCP כאשר כל המידע חשוב לנו הודעות, מיילים וניתן להותיר קצת איחור למען הגעת המידע כולו. UDP מועדף כאשר חשוב לנו המידע בזמן אמת כגון ZOOM שידורים או שיחת טלפון.**
- ח. (6 נקודות) מהו תפקיד פרוטוקול ה-DNS?
- לשלוח פקטות (frame) מחשבי קצה בתוך אותה רשת (LAN connectivity).
 - לתרגם כתובת IP לכתובת MAC.
 - לתרגם שם השרת לכתובת IP.**
 - לתרגם שם השרת לכתובת MAC.
 - לשלוח פקטות בין מחשבי קצה ברשתות שונות (WAN).
 - לאפשר תקשורת בין שני תהליכים במחשבי קצה ברשתות שונות (WAN).

נימוק: IP אינה קריא/שימושי לבני אדם, אנחנו משתמשים ב-host name כדי לגשת לאתר/מכונה מסוים/ת. משהו שמכונה לא מבינה, הפרוטוקול יוצר גשר בין אדם למכונה על ידי מיפוי ה-host name ל-IP.

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: כדי לפתור את הבעיה של המגבלות במספר ה-IP, ה-NAT משתמש ב-public IP אחד ברשת LAN מסוימת עבור מספר מכשירים באותו LAN מה שנותן re-use של ה-IP ADDRESSES.

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.

נימוק: כדי להתחיל את החיבור הלקוח צריך את ה-IP של השרת בשביל קריאת המערכת CONNECT ומספר ה-PORT ידוע, השרת גם צריך את ה-IP של הלקוח שהוא מקבל בקריאה ACCEPT כדי להשלים תהליך לחיצת היד המשולשת השרת לא צריך את ה-PORT של הלקוח מכיוון שהוא מתקשר איתו על ה-PORT שלו, לא צריכים את ה-DOMAIN מכיוון שהוא לא נחוץ לקריאת המערכת, ה-DOMAIN מתורגם בסוף ל-IP.

במודל שרת-לקוח הלקוח הוא זה שיוצר את התקשורת, ולכן עליו לדעת את הפרטים של השרת ולא להפך, כמו שכתבתם נכון בסעיפים הקודמים. את הפורט כמובן ששניהם צריכים לדעת! כמו שציננתם נכון בסעיפים הקודמים... אחרת הם לא יוכלו לתקשר.

6

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 מסוים 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>members++;</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:

a+d: בקוד הנתון הפונקציה `ONLEAVE` משחררת את המנעול לפני עדכון ה-`MEMBERS` לכן אם היה חדר ריק ואז נכנס מעריך של נוגה ושחרר את המנעול ומיד מערכת ההפעלה ביצעה החלפת הקשר לתהליך שמכניס מעריך של מרבי נקבל ששני אנשים מאותה קבוצה באותו חדר(בהפך למצב הרצוי מהקוד, כאשר הפונקציה הייתה מעדכנת השדה לפני החלפת ההקשר).

C. במקרה שאדם מקבוצת נוגה נכנס ו אחריו באו כמה אנשים מקבוצת מרבי שמנסים לנכנס אז החוטים שלהם יישארו תקועים ב `while` ולכן ניצלות המעבד נפגעת כאשר בשורה 13 בקוד בודקים תמיד את התנאי ואיננו מחקים לסיגנל.

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.

הפונקציה onleave שולחת סיגנל ככל אדם יוצא ממנה מה שיגרום לחוסים להתעורר גם אם החדר לא התפנה (אם היה 5 אנשים מקבוצה כלשהי בחדר ויצא אחד אז יישלח סיגנל לכל אלו שמהקבוצה השנייה והחוסים יתעוררו אבל תנאי הלולאה עדיין true וכולם יחזרו להמתנה שוב).

חסרה דוגמת ריצה קונקרטית
3-

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

להחליף את שורה 16 ב :

```
If(students[faculty] == 0)
    cond_signal(&conds[other]);
```

ואז הסיגנל נשלח רק כשהחדר מתפנה לכן החוסים לא יתעוררו סתם אלא יתעוררו רק אם החדר פנוי ועכשיו הם יכולים לכנס.

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[team] , &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 = members[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 למשאב 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:

a. נניח ש חוט 1 וחוט 2 הם מעריצי נוגה וחוט 3 הוא מעריץ מרבי עכשיו מגיעים שני מעריצי נוגה והדבר הבא קורה
חוט 1 מתחיל לרוץ בפונקציה onarrive עד שורה 9 ואז לפני שהוא מתחיל בשורה 10 (temp = 0) מתבצע context switch וחוט 2 מתחיל לרוץ עכשיו ומבצע onarrive (עכשיו members [team] = 1) עכשיו מעריץ מרבי מגיע וקורה context switch וחוט 3 מתחיל לרוץ בפונקציה onarrive אז הוא עוצר ב wait בגלל ש members [team] = 1 עכשיו קורה context

switch ו' חוט 2 חוזר לעבוד ומבצע onleave עכשיו $members[team] = 1$ ושולח סיגנל עכשיו קורה עוד context switch ו' חוט 3 עובד הוא קבל סיגנל ו' $members[team] = 1$ לכן הסטודנט יכנס עכשיו קורה context switch וחוט 1 ממשיך את ריצתו והסטודנט הראשון יכנס ואז יש לנו שני סטודנטים משני פקולטות שונות באותו חדר.

b: נכנס מעריץ של נוגה בחוט 1 מבצע פו onarrive לאחר מכן נכנס מעריץ של נוגה בחוט 2 הוא מגיע לשורה 9 מעדכן את $temp = 1$ ו' אז קורה context switch לחוט 1 שרוצה לצאת (לעשות פו onleave) ו' אז הוא מעדכן את $MEMBERS[1]$ לאפס ו' יוצא ו' קורה context switch לחוט 2 ו' הוא ממשיך משורה 9 בפו onarrive כך ש $temp$ אצלו שווה ל 1 ו' עכשיו הוא מעדכן את $MEMBERS[1]$ ל 2 למרות שיש מעריץ 1 בחדר ו' כשהוא יוצא יעדכן את $MEMBERS[1]$ ל 1 למרות שאין אף מעריץ בחדר ו' כשיבוא מעריץ מרבי שרוצה לנכנס לחדר הוא לא יכול כי $MEMBERS[1] = 1$.

d: נכנס מעריץ נוגה בחוט 1 מבצע פו onarrive לאחר מכן נכנס מעריץ מרבי בחוט 2 קורה context switch וחוט 2 מתחיל לעבוד בפו onarrive רץ עד שורה 7 שהתנאי בה הוא sure כי יש מעריץ נוגה בחדר ואז קורה context switch ו' חוט 1 רץ על הפו onleave שולח סיגנל אבל אין אף חוט בהמתנה עכשיו חוט 2 ממשיך לעבוד הוא מחכה לסיגנל אבל הסיגנל כבר נשלח והחדר ריק לכן הוא לא יקבל סיגנל למרות שהחדר ריק (כי הסיגנל כבר נשלח) ואז יש לנו סיגנל שנאבד ובגלל זה יש מעריץ שלא נכנס למרוט שהחדר ריק.