

Homework 3 Dry

Student	id	email
Amal hihi	213519333	Hihi.amal@campus.technion.ac.il
Sarah hamou	329618169	sarah@campus.technion.ac.il

Due Date: 12.1.2023

Teaching assistant in charge: Mousa Arraf

שאלה 1 - Networking - תקשורת (50 נק')

1. (4 נק') באיזו אבסטרקציה מרכזית של לינוקס משתמשים כאשר מתקשרים באמצעות ממשק ה-sockets?

Sockets הוא file-descriptors ולכן האבסטרקציה המרכזית בה משתמשים כאשר מתקשרים באמצעות ממשק ה-Sockets הוא "everything is a file".

2. (6 נק') ציינו את שתי הבעיות העיקריות בהעברת נתונים ברשת וכיצד מתמודדים איתן.

- בעיה 1: איבוד חלק מהמידע בדרך ליעד.
- בעיה 2: הגעה של המידע לא לפי הסדר.

כיצד מתמודדים איתן: ניתן להשתמש בפרוטוקול TCP המבטיח שמירת על הסדר המידע והגעתו. איך פרוטוקול ה-TCP

מונע את הבעיות האלו?

1. המקבל שולח ack- אישור שהוא קיבל את המידע וכך הפרוטוקול דואג שלא יאבד מידע- כי במידה והשולח לא מקבל אישור על כך שהמידע ששלח אכן הגיע למקבל עד זמן time-out מסוים, ישלח את המידע מחדש.
2. כל פאקטה נשלחת עם מספר סידורי שגדל עם כל פאקטה שנשלחת (ראינו בהרצאה שהמספר הזה הוא מספר הסידורי שידוע לשתי התהליכים שהתקבל ב-handshake + הגודל של כול הפאקטות שנשלחו עד כה). כשהמקבל מקבל פאקטות לא לפי הסדר, הוא יכול לזהות זאת על ידי המספר הסידורי של הפאקטה והמקבל ישמור את הפאקטות אבל לא יעביר את המידע לאפליקציה עד שיקבל את הפאקטות שהיו צריכות להתקבל פני הפאקטה הנוכחית ויסדר אותן בסדר המקורי. המקבל בסוף ישלח הודעה שקיבל את הפאקטות עד לביט x כלשהו התקבלו כלומר, שכול המידע עד x הגיע בשלמותו. וכך הפרוטוקול דואג שהמידע שיעבור לאפליקציה יעבור לפי הסדר הנכון (למרות שה-TCP לאוו דווקא יקבל אותו לפי הסדר!).

3. (12 נק') מלאו את הטבלה שלמטה כך שתסביר את ההבדל בין כתובת MAC, כתובת IP, ופורט.

כתובת MAC	כתובת IP	כתובת פורט
MAC Address	מספר זיהוי ברשת ה-LAN שהיא הרשת הלוקאלית.	כיצד הוא מתקבל הוא מתקבל בעת היצירה של כרטיס הרשת כאשר המספר מורכב מ-48 ביטים שחלקם הם מאפיינים את יצרן הכרטיס וחלק פרטי וייחודי לכרטיס רשת הספציפי.

<p>כתובת ה- IP מתקבלת על ידי פרוטוקול ה- DHCP פרוטוקול המשמש להקצאה אוטומטית של כתובת IP עבור מכשירים שלא הוגדרה להם מראש כתובת IP סטטית מראש ברשת המקומית.</p>	<p>לכל כרטיס רשת יש כתובת IP שהיא רצף של 4 בתים המוקצית לו ומשמשת לזיהוי שלו ברשת.</p>	<p>IP Address</p>
<p>מוקצה באופן דינאמי דינאמי על ידי הקרנל פרט ל- Ports מוכרים.</p>	<p>מזהה של תהליך במחשב שמנהל ערוץ תקשורת (*) בין תהליכים. *socket</p>	<p>Port</p>

4. (20 נק')

א. ווילם, מנכ"ל של חברת הייטק נחשבת, חושש שספקית האינטרנט שלו תאזין לתקשורת שלו ותמכור מידע רגיש למתחרים ולכן הוא שוקל להצפין את ה-payload של שכבה 2 אשר המחשב שלו משדר (מבלי לתאם את הדבר עם אף גורם). באיזה שלב בדרכה של החבילה אל היעד ניתקל בבעיה לראשונה?

• נפריד למקרים:

מקרה א':

המחשב איתו ווליום מנסה לתקשר לא נמצא ברשת הפנימית שלו, כלומר נצטרך לעבור דרך הראוטר, אז אתקל בבעיה הראשונה בראוטר, שלא יוכל לפענח את ה-ip של היעד זאת משום שה-ip נמצא ב-packet header שמוצפן.

מקרה ב':

המחשב איתו ווליום מנסה לתקשר נמצא ברשת הפנימית שלו, ואז הבעיה הראשונה תיווצר במחשב היעד, שם לא נוכל לפענח לאיזה תהליך שייך המידע כיוון שה-port מוצפן כי הוא נמצא ב-segment-header.

ב. אם ווילם היה מצפין את ה-payload של שכבה 3 (במקום של 2), באיזה שלב בדרך של החבילה אל יעדה ניתקל בבעיה לראשונה?

הבעיה הראשונה הייתה נוצרת במכשיר היעד, כיוון שכול שכבה "מקלפת" את המידע שהיא זקוקה לו, וכאשר ננסה לקרוא את ה-port לא נוכל לפענח את המידע כיוון שהוא מוצפן, (אם מכשיר היעד מסוגל לפענח את המטען, ההצפנה לא אמורה לגרום לבעיות).

5. (8 נק') לוויילם יש קופסא של בזק בבית שמשמשת בשני מנגנונים עיקריים המאפשרים לכל התקן פרטי ברשת הביתית שלו לקבל כתובת IP שמאפשרת תקשורת אינטרנט. תארו את שני המנגנונים בקצרה.

פרוטוקול DHCP-

פרוטוקול תקשורת שמקצה באופן אקראי כתובות IP ייחודיות למחשבים שלא הוקצתה להם כתובת IP באופן סטטי, עבור מחשבים ברשת המקומית (LAN).

מנגנון NAT –

טכניקת ניתוב מחשבים ברשת, שממירה כתובת IP של רשת פרטית לכתובת IP ציבורית שיכולה לתקשר עם הרשת החיצונית. לבסוף, המנגנון יודע להחזיר את המידע מהרשת הציבורית שקיבל חזרה למחשב המבוקש.
(ה-IP והפורט מתורגמים כאילו נשלחו מהראוטר והפורט החדש משמש כאינדקס לטבלת ה-NAT - שעל ידיו הראוטר ידע לאיזה IP ברשת הפנימית הבקשה שייכת – קצת אקסטרה שקראתי בגוגל 😊).

שאלה 2 - סינכרון (50 נק')

המצאת המושג "פקולטה נחשבת" החמירה את הסכסוך בין הסטודנטים במדמ"ח ובהנדסת חשמל, ולכן הוגדר כי כאשר סטודנט מאחת הפקולטות רוצה להיכנס לחדר מסויים עליו לציית לכלל הבא: אם יש סטודנטים מפקולטה אחרת בחדר אזי אסור לסטודנט להיכנס ועליו להמתין עד שיעזבו (לעומת זאת, מספר סטודנטים מאותה פקולטה יכולים לשהות בחדר באותו הזמן).

סמני נכון / לא נכון (אין צורך להסביר):

1. (3 נק') יכולים להיות שני סטודנטים מפקולטות שונות באותו חדר במקביל: **נכון** / **לא נכון**
2. (3 נק') יכולים להיות שני סטודנטים מפקולטות זהות בחדר במקביל: **נכון** / **לא נכון**
3. (3 נק') סטודנטי פקולטה אחת עלולים להרעיב (כניסת) סטודנטי פקולטה אחרת: **נכון** / **לא נכון**

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

- כל חוט מייצג סטודנט.
- בכניסה לחדר הסטודנט קורא ל `onArrival(int faculty)`, שמקבלת את פקולטת הסטודנט.
- ביציאה מהחדר הסטודנט קורא ל `onLeave(int faculty)` שמקבלת את פקולטת הסטודנט.
- הערכים 0 ו-1 של `faculty` מייצגים את הפקולטה להנדסת חשמל ומדמ"ח, 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 faculty) {</code>
2. <code>int students = 0;</code>	12. <code>mutex_lock(&global);</code>
3. <code>mutex_t global;</code>	13. <code>while (students > 0) {</code>
4. <code>void onLeave(int faculty) {</code>	14. <code>mutex_unlock(&global);</code>
5. <code>mutex_lock(&global);</code>	15. <code>sleep(10);</code>
6. <code>students--;</code>	16. <code>mutex_lock(&global);</code>
7. <code>mutex_unlock(&global);</code>	17. <code>}</code>
8. <code>}</code>	18. <code>students++;</code>
9. <code>}</code>	19. <code>mutex_unlock(&global);</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:

C הקוד משתמש ב-Busy Wait שפוגע בנצילות המעבד :

הסטודנט הראשון נכנס students כבר לא 0, אחר מכן סטודנט 2 מנסה לכנס לחדר, אבל בשורה 13 הוא נתקע בלולאה while, כי ה condition (students > 0 == true) ואחר כך הוא מבצע בשורה 16, sleep(10), הפעולה הזאת משתמשת במעבד ולא גורמת לתהליך של second student לצאת להמתנה עד ש first student יוצא, ולכן כאן מתבצע פגיעה בנצילות המעבד.

```
//CS student enter the room
OnArrival(1);
//also CS student entering the room
OnArrival(1);
```

d הקוד מפר את כלל הכניסה לחדר (שהוגדר בתחילת השאלה):

```
///CS student enters the room
OnArrival(1);
///also CS student tries entering the room but fails because of line 13 student > 0
OnArrival(1);
```

- הקוד מפר את כלל הכניסה לחדר מכיוון שבהרצה הזו לא נאפשר לכך שיותר מסטודנט אחד ישהה בחדר ת למרות שהם מאותה פקולטה של סטודנט ראשון.

המימוש של כניסה ויציאה שונה כך שישתמש במשתני תנאי:

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

אך דני (עתודאי במדמ"ח) טען שקוד זה גורם לחוסים להתעורר שלא לצורך ומיד לחזור למצב המתנה.
1. (8 נק') הסבירי את טענתו של דני באמצעות דוגמת ריצה קונקרטית.

```
/*CS student 1 enters the room*/
OnArrival(1);
/*CS student 2 enters the room*/
OnArrival(1);
/*EE student 3 try to enter the room + wait */
OnArrival(0);
/*CS student 1 leaves the room*/
OnLeave(1);
```

The `pthread_cond_broadcast()` function shall unblock all threads currently blocked on the specified condition variable `cond`.

דני צודק, בדוגמה הרצה לעיל לאחר שסטודנט ממדמ"ח יעזוב את החדר, עדיין יש עוד יש סטודנט ממדמ"ח שנמצא בחדר, ולכן הסטודנט מס 3 לא יכול להיכנס לחדר, אך שסטודנט 1 עוזב מבצעים `cond_broadcast` המעירה את כל החוטים של הסטודנטים וגם הסטודנט מחשמל, אבל סטודנט זה יתעורר אבל יגלה שעדיין לא יכול להיכנס כי עדיין החדר מכיל מדמחיסטים, ויצא שוב להמתנה, וכאן נראתה טענת דני.

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

```
12 void onLeave(int faculty) {
13     mutex_lock(&global);
14     students[faculty]--;
15     int other = faculty ? 0 : 1;
16     cond_broadcast(&conds[other]);
17     mutex_unlock(&global);
18 }
```

בשורה 16 צריך לתקן:

לפני שעושים `cond_broadcast(&cond[other])`

נבדוק `if(students[faculty]==0)` ואז נבצע `broadcast`

עכשיו נעיר את הסטודנטים מהפקולטה השנייה, אם החדר ריק והסטודנט מהפקולטה האחרת עזב את החדר.

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

```
1 int students[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 faculty){
5     mutex_lock(&m_arrival);
6     int other = faculty ? 0 : 1;
7     while(students[other] > 0)
8         cond_wait(&conds[faculty], &m_arrival);
9     int tmp = students[faculty];
10    students[faculty] = tmp + 1;
11    mutex_unlock(&m_arrival);
12 }
13 void onLeave(int faculty){
```

14	mutex_lock(&m_leave);
15	int tmp = students[faculty];
16	students[faculty] = tmp - 1;
17	int other = faculty ? 0 : 1;
18	cond_broadcast(&conds[other]);
19	mutex_unlock(&m_leave);
20	}

1. (13 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 סטודנטים מפקולטות שונות בתוך החדר ביחד, עקב race condition למשאב משותף.

```
/*CS student 1 enters the room*/  
OnArrival(1);  
  
/*CS student leaves the room  
tmp=1 / line 15 after that context switch for another cs student*/  
OnLeave(1);  
  
/*CS student 2 enters the room  
student[faculty]=tmp+1=1+1=2 \\line 10  
untill now the value in the array have not changed */  
OnArrival(1);  
  
/*CS student leaves the room , context switch again ,  
student[faculty]==0 */  
OnLeave(1);  
  
/*EE student 3 enter the room because student[faculty]=0 */  
OnArrival(0);
```

מכיוון שיש גישה למשאב משותף המערך students משתי פונקציות שונות, אשר משתמשים במנעולים שונים, נוצרה התנהגות לא צפויה והזנת ערכים לא צפויים למערך, ולכן הראנו בדוגמה שאפשר לקבל מספר סטודנטים ממדמ"ח הינו אפס למרות שזה לא, ולאפשר לסטודנט מחשמל להיכנס בחדר בהצלחה.

b. יתכן סטודנט שלא נכנס לחדר למרות כלל הכניסה שמתיר זאת, עקב race condition למשאב משותף.

```
/*CS student 1 enters the room*/
OnArrival(1);
/*CS student enters the room
tmp=1 \ line 9 context switch after this line */
OnArrival(1);
/*CS student leaves the room
student[faculty]= tmp-1=1-1=0
untill now the value in the array did not change*/
OnLeave(1);
/*back to the CS student from context switch back to OnArrival
student[faculty]=tmp+1=1+1=2 \\line 10 */
OnArrival(1);
/*CS student 2 leaves the room */
OnLeave(1);
/*EE student 3 can't enter the room
because student[faculty]=1 even though it's empty */
OnArrival(0);
```

מכיוון שיש גישה למשאב משותף משתי פונקציות שונות (מערך סטודינטיים) , אשר לכל אחת מנעול בנפרד , נגרמת התנהגות לא צפויה , וערכים לא צפויים במערך , בדוגמה הראנו איך אפשר לקבל במערך הערך 1 למרות שאין אף סטודנט ממדמ"ח בחדר , וכך נגרם לסטודנט מחשמל אי אפשרות לכנס לחדר למרות שאין אף אחד .

c. קיימת בעיית DeadLock / Livelock בקוד. \\ לא מתקיים

d. סיגנלים עלולים ללכת לאיבוד.

```
/*CS student 1 enter the room*/  
OnArrival(1);  
/*EE student 2 enter the room  
context switch after line 7 */  
OnArrival(0);  
/*CS student leaves the room*/  
OnLeave(1);  
/*back to OnArival for EE student  
line 8 'cond_wait(&conds[faculty] , &m_arrival)"/  
OnArrival(0);
```

סטודנט ממדמ"ח נכנס לחדר, סטודנט מחשמל מנסה לכנס, מתבצעת החלפת הקשר בשורה 7 לפני שהוא נכנס להמתנה ל signal של סטודנט ממדמ"ח יצא מהחדר ואז הוא יכול להיכנס, ולכן משדר לכל הסטודנטים מחשמל שהוא יצא אבל הסטודנט מחשמל עדיין לא חיכה לסיגנל הזה, ולכן הסטודנט המסכן הזה מחשמל ימשיך לחכות לסיגנל למרות שאין אף אחד בחדר והוא יכול להיכנס.