Homework 3 Dry

**Due Date: 04.08.2024**

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:
   1. The first page should contain the details about the submitters - Name, ID number and email address.
   2. 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 1 - Networking - תקשורת**

לאחר חוסר הצלחת הפיצריות של המתרגלים בפקולטה, החליטו לפנות אל האינטרנט במטרה לפרסם את בתי העסק שלהם. לצערנו, אין המתרגלים יודעים כיצד עובד האינטרנט, אנא עזרו להם לענות על השאלות הבאות.

**חלק פתוח:** הסבירו בקצרה (לא יותר מ1-2 שורות!).

**א.** הסבירו מה תפקיד של פרוטוקול ARP.

תשובה:  
פרוטוקול APR (Address Resolution Protocol) הינו פרוטוקול אשר בעזרתו ניתן לגלות את כתובת ה- MAC של כל מארח באותה רשת מקומית באמצעות כתובת הIP שלו מכיוון שהמידע משודר לפי כתובת MAC של היעד ברמה הפיזית.

**ב.** הסבירו את משמעות המושג .

תשובה:  
default gateway הוא הנקודה (node) שמשמשת כנתב (router) להעברת חבילות נתונים לרשתות אחרות כאשר אין הגדרה אחרת שמתאימה לכתובת ה-IP של היעד. כלומר, אם המחשב לא יודע לאן לשלוח את החבילה בתוך הרשת המקומית, הוא ישלח אותה ל-default gateway, שידאג להעביר אותה הלאה.

**ג.** איזה מידע צריך לדעת הלקוח על השרת לפני ההתחברות?

תשובה:  
על הלקוח לדעת שלושה דברים:

* 1. על הלקוח לדעת מה ה-port שאליו האפליקציה מזינה
  2. על הלקוח לדעת מה כתובת הIP שלו על מנת שיוכל למצוא את הכתובת הנ"ל ואת כתובת הMAC שלו ברשת.
  3. סוג הפרוטוקול

**ד.** איזה מידע השרת צריך לדעת על הלקוח לפני ההתחברות?

תשובה:  
השרת לא צריך לדעת כלום על הלקוח זאת מאחר ותפקיד השרת הוא להיות נגיש עבור כל לקוח אשר פונה אליו ולכן אין הוא יכול לדעת איזה מידע הוא יקבל.

**ה.** כיצד שרת יודע את כתובת ה- של הלקוח?

תשובה:  
כתובת ה-IP של הלקוח נכללת באופן אוטומטי בתקשורת ברשת, והשרתים מתוכננים בהתאם לסוג הרשת והפרוטוקולים שבהם משתמשים לזהות ולנתח את המידע הזה לצורך ניהול נכון של התקשורת.

**ו.** מה הבדל בין הפורט (port) שבשימוש השרת וזה של הלקוח. איך נבחר כל אחד מהם?

תשובה:  
הפורט בשימוש השרת והפורט בשימוש הלקוח נבדלים בתפקידיהם ובאופן בחירתם. הפורטים של השרת הם בדרך כלל פורטים קבועים וידועים אשר נבחרים על פי הפרוטוקול והשירות שהשרת מספק. פורטים אלו מוגדרים מראש ועל כן הלקוחות חייבים לדעת לאיזה פורט לפנות כאשר הם מבקשים שירות מסוים.מנגד, הפורטים של הלקוח הם דינמיים ונבחרים באופן אוטומטי על ידי מערכת ההפעלה של הלקוח במהלך יצירת הsocket.

**ז.** מה הבדל בין פרוטוקול TCP ו-UDP? הסבירו למה חלק מהאפליקציות מעדיפות TCP וחלק UDP.

תשובה:  
ההבדל בין פרוטוקול TCP והפרוטוקול UPD נמדד בעיקר באמינות כל פרוטוקול, פרוטוקול TCP הינו פרוטוקול אמין אשר יוצר תקשורת דו כיוונית בין הלקוח לשרת ומוודא באמצעות אישורים מן הצד השני כי המידע הגיע ובמידה והמידע הולך לאיבוד מטפל במקרה.  
לעומתו פרוטוקול UDP אינו פרוטוקול אמין ובעצם לא יוצר קישור ולא מוודא את הגעת המידע אל הצד השני ובעקבות כך מידע יכול להיאבד.  
בעקבות ההבדלים הללו נוצר הבדל של מהירות. מאחר ופרוטוקול TCP מוודא את הגעת המידע דבר אשר כרוך גם במידת הצורך בשליחה נוספת של המידע נוצר מצב ובו הפרוטוקול TCP איטי יותר מן הפרוטוקול UDP אשר שולח את המידע פעם אחת ולא מוודא את הגעתו.  
כל אפליקציה מבצעת את עדיפות האמינות אל מול מהירות שלה כדוגמא אפליקציה אשר שולחת מידע חשוב למקום אחר תעדיף כי המידע יגיע גם אם הוא יגיע באיטיות יותר מביצוע פרוטוקול ובו ישנו הסיכוי והמידע לא יגיע.  
לעומת זאת אפליקציה אשר באמצעותה אנו יכולים לראות סרט באינטרנט תעדיף כי הצפייה עבור הצופה תהיה רציפה ועל כן תעדיף פרוטוקול מהיר יותר.

**ח.** האם ייתכן שמחשב לדעת האם מחשב היעד נמצא איתו באותה הרשת ללא שליחת הודעות בהינתן כתובת ה- שלו?

תשובה:  
אנו נוכל לבצע את הפעולה הבינארית AND בין כתובת היעד אל לבין מסכת הרשת (Subnet Mask). התוצאה תהיה כתובת הרשת של היעד.   
אם כתובת הרשת שלך וכתובת הרשת של היעד זהות, אזי המחשב היעד נמצא באותה הרשת המקומית. אם הן שונות, אז המחשב היעד נמצא ברשת אחרת.

**ט.** מהו תפקידו של פרוטוקול DNS?  
תשובה:  
פרוטוקול DNS בעצם מתרגם את שם השרת (domain) לכתוב IP. שם השרת הוא עבור המשתמש על מנת שהמשתמש לא יהיה זקוק לזכור את כתובות הIP, אך מאחר ומחשבים מתקשרים אך ורק בביטים ולא באותיות אזי הפרוטוקול מתרגם את שם השרת לכתובת הIP הנכונה ובכך הממשק נוח יותר עבור המשתמש.

**י.** כיצד מערכת ההפעלה יודעת לאיזה תהליך מיועדת פקטה נכנסת?

תשובה:

כאשר פקטה נכנסת מגיעה למחשב, מערכת ההפעלה משתמשת בכתובת ה-IP ופורט היעד שבכותרת הפקטה כדי לנתב אותה לתהליך הנכון. מערכת ההפעלה מחזיקה טבלת פורטים פעילים (Sockets), שבה כל חיבור פתוח מקושר לכתובת IP ופורט יעד מסוימים. כשהפקטה מתקבלת, מערכת ההפעלה בודקת את פורט היעד שלה מול הטבלה הזו. אם נמצאה התאמה, הפקטה נשלחת לתהליך המתאים שמאזין על אותו פורט. בצורה זו, הפקטה מגיעה ליישום או השירות הנכון שמחכה לה.

**יא.** איזה מידע מספק פרוטוקול למחשב המתחבר לרשת?

תשובה:  
פרוטוקול DHCP מספק למחשב המתחבר לרשת מידע חיוני לתפקודו. הוא מעניק למחשב כתובת IP ייחודית, מסכת רשת (Subnet Mask), וכתובת Default Gateway . בנוסף, הפרוטוקול מספק כתובות של שרתי DNS לתרגום שמות דומיין לכתובות IP ואת משך השכירות של הכתובת (Lease Time) כל זה מאפשר למחשבים להתחבר לרשת בקלות ובאופן אוטומטי, ללא צורך בהגדרות ידניות.

**יב.** 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**:   
בפרוטוקול IPv4 כתובת ה-IP מורכבת מ-4 מספרים בני 8 ביטים כל אחד. לכל כתובת יש שני סוגים (מאחר ואין מספיק כתובות על מנת שתהיה האפשרות לייצר כתובות ייחודיות לכל מכשיר אינטרנטי):   
כתובת IP גלובלית המיועדת לראוטר וכתובת מקומית בשרת (LAN) שאינה גלובלית.  
בשימוש ב-NAT כתובות IP מקומיות מומרות לכתובת IP גלובלית אחת, כאשר אנו רוצים להמיר חזרה את הכתובת לכתובת מקומית, אנו משתמשים במספרי הפורטים (ports) כדי לנתב את המידע למכשיר המתאים ברשת המקומית.

**יג.** 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**:

השרת כפי שנאמר גם למעלה לא צריך לדעת כלום על הלקוח לפניי החיבור על מנת שכל לקוח יוכל להתחבר אליו ועל כן בעקבות כך על הלקוח לדעת בדיוק לאיזה שרת הוא מעוניין להתחבר ולכן על הלקוח לדעת את שם הdomain על מנת שיוכל להמיר אותה אל כתובת IP, ובנוסף עליו לדעת את מספר הפורט שאליו השרת מאזין. בצורה זו, הלקוח מסוגל להתחבר לשרת הנכון ולשירות הנכון בתוכו, מה שמבטיח תקשורת תקינה בין הלקוח לשרת.

שאלה 2 - סינכרון (48 נק')

לאחר תרגיל בית 1 במת״מ החליטו המתרגלים לעבור להשתמש בתור רגיל, שאינו ישראלי, אך כעת ישנם מספר חוטים שניגשים אליו במקביל ולכן עליכם לעזור להם לדאוג כי ניתן יהיה לגשת אל התור במקביל בצורה בטוחה.

1. לפניכם מימוש חלקי של התור, השלימו את פונקציות ההוצאה וההכנסה לתור כך שיתמכו בצורת ריצה מקבילית.

* ניתן להניח כי קיימות הפונקציות וניתן להשתמש בהן באופן חופשי.
* לתור יש גודל מקסימלי המוגדר בקוד.

|  |  |  |  |
| --- | --- | --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21. | struct SyncQ{  Queue queue;  mutex\_t \*lock;  cond\_t \*has\_room;  cond\_t \*has\_elements;  }  #define MAX\_SIZE 10  Req\* sync\_pop(SyncQ \*q){  mutex\_lock(q->lock);  while(q->queue.size()==0){  cond\_wait(q->has\_element , q->lock);  }  Req\* retVal = q->queue.pop();  cond\_signal(q->has\_room);  mutex\_unlock(q->lock);  return retVal;  } | 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21. | void sync\_push(SyncQ \*q, Req \*r){  Mutex\_lock(q->lock);  while(q-> queue.size()>=MAX\_SIZE){  cond\_wait(q->has\_room , q->lock);  }  q-> queue.push(r);  cond\_signal(q->has\_elements);  mutex\_unlock(q->lock);  } |

כעת לאחר שיש בידם תור שעובד בגישה מקבילית, לא נחו המתרגלים והחליטו כי עליהם לכתוב מנעול קוראים כותבים המקיים את תכונות mutex. מספר חוטים קוראים יחדיו יכולים להחזיק במנעול, אך כאשר חוט כותב תופס את המנעול, עליו להיות היחיד. אם יש חוט אשר מנסה לתפוס את המנעול, והוא פנוי, עליו להצליח מיד.

|  |  |  |  |
| --- | --- | --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10. | #include <pthread.h>  int readers\_inside = 0;  int writers\_inside = 0;  mutex\_t global;  void **read\_unlock**() {  readers\_inside --;  } | 11.  12.  13.  14.  15.  16.  17.  18.  19.  20. | void **read\_lock**() {  mutex\_lock(&global);  while (writers\_inside > 0) {  mutex\_unlock(&global);  sleep(10);  mutex\_lock(&global);  }  Readers\_inside++;  mutex\_unlock(&global);  } |

|  |  |  |  |
| --- | --- | --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10. | void **write\_unlock**() {  writers\_inside --;  } | 11.  12.  13.  14.  15.  16.  17.  18.  19.  20. | void **write\_lock**(int team) {  mutex\_lock(&global);  while (readers\_inside > 0) {  mutex\_unlock(&global);  sleep(5);  mutex\_lock(&global);  }  Writers\_inside++;  mutex\_unlock(&global);  } |

1. בהתייחס לקוד הנ״ל, ציינו את 3 בעיות הנכונות והיעילות הקיימות בקוד, עבור כל בעיית נכונות תארו את תרחיש הריצה הבעייתי.

נימוק:

בעיה ראשונה – בעיית יעילות:  
גם עבור write\_lock וגם עבור read\_lock כאשר אנו נותנים אפשרות קריאה או כתיבה אנו מעניקים זאת באמצעות אינטרוול זמן קבוע (sleep(5) ו- sleep(10)) ועל כן יכולה להיווצר סיטואציה ובה אנו קוטעים פעולה באמצע או ממתינים זמן רב מדי למרות שהפעולה כבר הסתיימה ממזמן.  
תרחיש הריצה הבעייתי:

1. חוט 1 נכנס לכתוב
2. חוט 2 נכנס לקרוא, רואה כי ישנו כותב ועל כן מחכה 10 יחידות זמן
3. חוט 1 מסיים לכתוב לאחד יחידת זמן 1
4. חוט 2 נכנס לקרוא רק לאחר 9 יחידות זמן נוספות מאחר וכך הוגדרה הפעולה

בתרחיש הנ"ל אנו יכולים לשים לב כי זוהי לא דרך פעולה יעילה מאחר ובמקום שלאחר יחידת זמן אחת חוט 2 יוכל לקרוא עליו להמתין 9 יחידות זמן "מיותרות" נוספות על מנת להתחיל ולקרוא מהקובץ.

בעיה שנייה – בעיית נכונות:

עבור קטעי הקוד read\_unlock וwrite\_unlock כאשר אנו רוצים להוריד את מספר הכותבים או מספר הקוראים בקובץ עלינו לעטוף את הפעולה במנעול כפי שראינו בהרצאה ובתרגול. זאת על מנת שלא ייקרה מצב ובו שני קוראים או שני כותבים מורידים בהפרש זמנים קטן את מספר הכותבים או הקוראים בקובץ ובכך נוצר מצב ובו אנו לא יודעים את המספר לא מתעדכן בצורה נכונה.

תרחיש הריצה הבעייתי:

1. חוט 1 נכנס לקרוא
2. חוט 2 נכנס לקרוא באינטרוול זמן זניח ועל כן מעלה את כמות הקוראים מ1 ל2
3. חוט 1 יוצא ומוריד את כמות הכותבים.
4. חוט 2 יוצא גם הוא ומוריד בו זמנית את מספר הקוראים

כעת אנו נמצאים במצב שאנו לא יודעים מהו מספר הקוראים בקובץ מאחר והורדת כמות הקוראים התבצעה בו זמנית.

בעיה שלישית – בעיית נכונות:

עבור קטע הקוד בwrite\_lock כאשר אנו בודקים האם חוט יכול לכתוב אל הקובץ אנחנו מוודאים רק האם לא קיימים קוראים בקובץ ולא בודקים האם הוא החוט הכותב היחידי דבר אשר יכול לגרור למצב ובו ישנם שני חוטים כותבים בניגוד לתנאי השאלה.

תרחיש הריצה הבעייתי:

1. חוט 1 נכנס לכתיבה מוודא כי אין חוטים נוספים אשר קוראים ונכנס.
2. חוט 2 נכנס לכתיבה מוודא גם הוא כי אין חוטים נוספים אשר קוראים ונכנס גם הוא.
3. שני החוטים כותבים בו זמנית לאותו הקובץ

בתרחיש זה ניתן לשים לב כי ישנו מצב ושני חוטים כותבים בו זמנית בניגוד לדרישות השאלה

כעת הבין סגל מת״מ כי משתני תנאי חזרו לאופנה ולכן רצו לשנות את מימוש המנעול שלהם כך שישתמש בכאלו.

לפניכם המימוש המחודש:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | Int readers\_inside = 0;  Int writers\_inside = 0 ;  cond\_t conds\_writer;  cond\_t conds\_reader;  mutex\_t global;  void **reader\_lock**() {  mutex\_lock(&global);  while(writers\_inside > 0)  cond\_wait(&conds\_reader, &global);  readers\_inside++;  mutex\_unlock(&global);  }  void **reader\_unlock**() {  mutex\_lock(&global);  readers\_inside--;  cond\_broadcast(&conds\_writer);  mutex\_unlock(&global);  }  void **writer\_lock**() {  mutex\_lock(&global);  while(readers\_inside+writers\_inside > 0)  cond\_wait(&conds\_writer , &global);  writers\_inside++;  mutex\_unlock(&global);  }  void **writer\_unlock**() {  mutex\_lock(&global);  writers\_inside--;  cond\_broadcast(&conds\_readers);  mutex\_unlock(&global);  } |

אך לאחר שלסטודנטים נמאס מרעיונותיו של הסגל, עדי הגיע להציל את המצב וכשראה את הקוד טען שקוד זה גורם לחוטים להתעורר שלא לצורך ומיד לחזור למצב המתנה.

1. הסבירי את טענתו של עדי באמצעות דוגמת ריצה קונקרטית.

תשובה:

דוגמת הריצה הבעייתי:

* 1. **חוט 1 נכנס לקרוא**
  2. **חוט 2 נכנס לקרוא**
  3. **חוט 3 נכנס לכתוב**
  4. **חוט 3 ממתין למנעול**
  5. **חוט 1 מסיים לקרוא**
  6. **חוט 3 מתעורר בעקבות הסיגנל אשר מעיר את חוט 3 ומודיע לו על הורדת כותב אך חוט 3 חוזר להמתנה מאחר ויש קורא נוסף**

**בדוגמא הנ"ל ראינו כי חוט 3 מתעורר שלא לצורך וגורר בעיית יעילות**

1. כיצד ניתן לתקן את הבעיה שהציג נדב בסעיף הקודם?

**תשובה:**

**אנו יכולים לתקן את הבעיה באמצעות פעולת if אשר תבדוק לפניי שליחת הסיגנל כי אין קוראים נוספים כלל ואנו במצב של 0 קוראים (מצב ובו ניתן לכתוב אל הקובץ).**

1. כעת לאחר התיקון, טען עדי כי הקוד תקין. האם עדי צודק? אם כן, הסבירו מדוע. אם לא, ספקו דוגמת הרצה אשר בה המנעול לא מתנהג כמו שצריך.

נימוק:

**הקוד עדיין אינו תקין זאת מאחר ואנו יכולים להגיע למצב ובו יש "הרעבה" עבור הכותבים זאת מאחר ויכולים להמשיך ולהיכנס קוראים ובכך נגיע למצב כי הכותבים אף פעם לא יכולים לגשת ולכתוב לקובץ.**