

### מערכות הפעלה - תרגיל בית יבש 3

**Ido Tausi**, 214008997, ido.tausi@campus.technion.ac.il

**Noam Bitton**, 213745953, bitton.noam@campus.technion.ac.il

## שאלה 2 - תקשורת

א. הסבירו מה תפקיד של פרוטוקול ARP.

תפקידו של פרוטוקול ARP הוא לאתר כתובת Mac (כתובת פיזית) של תחנה ברשת המקומית לפי כתובת ה-IP שלה, כדי שיוכלו לשלוח מידע ביניהם (מתרגם בין כתובת IP לבין כתובת MAC).

ב. איזה מידע הלקוח צריך לדעת על השרת לפני ההתחברות?

הלקוח צריך לדעת את כתובת השרת (IP) ואת מספר ה-*port* שהשרת מאזין לו. ייתכן מצב בו נדע בהתחלה רק את ה-*hostname* של השרת, ולאחר שניעזר ב-DNS נקבל את כתובת השרת.

ג. איזה מידע הלקוח ידע על השרת אחרי ההתחברות?

לאחר ההתחברות, הלקוח יידע על השרת את כל המידע שידע לפני ההתחברות כלומר ה-IP והפורט של השרת.

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

לפני ההתחברות השרת לא צריך לדעת דבר על הלקוח. השרת מאזין לפורט מסוים ומחכה לקבלת בקשה, כאשר נשלחת בקשה מלקוח אל הפורט אליו השרת האזין, השרת מאשר את הבקשה ומבצע התחברות.

ה. איזה מידע השרת ידע על הלקוח אחרי ההתחברות?

לאחר ההתחברות השרת ידע את ה-IP ואת הפורט של הלקוח.

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

ההבדל הוא שהפורט שבשימוש השרת תמיד רץ ומקשיב לבקשות שיגיעו אליו, ומספרו קבוע וידוע ללקוח מראש. לעומת הפורט האקראי שבשימוש הלקוח שיהיה פעיל רק כאשר יש לנו צורך בתקשורת מול השרת.

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

ההבדל הוא ש-TCP הוא פרוטוקול המבטיח תקשורת אמינה בין הצדדים, כלומר מאפשר העברת פקטות כך שכל המידע יגיע בשלמותו ובסדר הנכון בעוד UDP הוא פרוטוקול תקשורת לא אמינה. אפליקציות שמעדיפות השהייה נמוכה על פני אמינות יעדיפו UDP ואילו כל השאר יעדיפו TCP. בנוסף, פרוטוקול TCP דואג לסדר הפקטות שהוא מעביר לאפליקציה לפי סדר שליחתן ואישור קבלתן לשרת, בניגוד לפרוטוקול UDP שמעביר הלאה כל פקטה שהוא מקבל ללא אישור קבלה לשרת. הבדל נוסף הוא הצורך בייסוד החיבור. בעוד שפרוטוקול TCP מבוסס חיבור, ב-UDP התקשורת מתחילה מיד ללא צורך בייסוד חיבור.

ח. מהו תפקיד פרוטוקול ה-DNS?

תשובה ג': לתרגם שם השרת לכתובת IP. פרוטוקול ה-DNS, או *Domain Name Server* ממפה את השמות איתם מזוהים השרתים (אלו הם שמות הניתנים לצורך נוחות בני האדם) לכתובות IP כדי שניתן יהיה לתקשר איתם ברשת האינטרנט.

ט. מהו תפקיד פרוטוקול ה-NAT?

תשובה ב': שימוש של מספר קטן של כתובות IP עבור הרבה מכשירים בתוך הרשת. מספר כתובות ה-IPv4 מוגבל, ולכן לא ניתן לתת כתובת IP ייחודית לכל מכשיר בעולם. לכן תהיה הפרדה בין רשת פנימית לחיצונית על ידי הראוטר, כאשר ברשת הפנימית התקשורת תעשה על ידי כתובת IP פנימית כלשהי ותקשורת מבחוץ על ידי הכתובת החיצונית, דרך הראוטר כלומר כל המחשבים ברשת הפנימית מיוצגים על ידי אותה כתובת חיצונית. תפקיד פרוטוקול NAT הוא להמיר את הכתובת הפנימית של לקוח ברשת, לכתובת חיצונית עם פורט בראוטר, וכך יוכל לבסס תקשורת בין לקוח לשרת. פורט זה חשוב מפני שכך הראוטר יידע לנווט את המידע שיתקבל בפורט זה, אל המחשב ברשת הפנימית.

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

תשובה א': במודל תקשורת זה, השרת לא חייב לדעת דבר על הלקוח, והלקוח צריך לדעת מה שם הדומיין של השרת על מנת להמירו לכתובת IP, וגם מספר פורט בשרת שידוע לקבל בקשות מסוג זה.

## שאלה 2 - סינכרון

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

1. בהתייחס לקוד הני"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 condition race למשאבים משותפים.**

b. קיימת בעיית Livelock / DeadLock בקוד.

**c. הקוד משתמש ב- *Busy Wait* שפוגע בנצילות המעבד.**

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

דוגמת הרצה לתשובה a:

נניח כי בחוט אחד מתבצע  $OnArrival(0)$ , ברגע זה המנעול פנוי ולכן המנעול יינעל. כעת  $members = 0$  ולכן לולאת ה- $while$  לא תתבצע והמנעול ייפתח. נניח כי מיד לאחר פתיחת המנעול בשורה 18, תתבצע החלפת הקשר לחוט שמבצע  $OnArrival(1)$ , ברגע זה המנעול פנוי, לכן המנעול יינעל, וכעת  $members = 0$  ולולאת ה- $while$  לא תתבצע והמנעול ייפתח. כעת אנו נבצע  $members++$ , ויתקיים  $members = 1$ . בנוסף, ברגע שנחזור לחוט הראשון, נמשיך מהנקודה שעצרנו, כלומר נבצע  $members++$  ולכן כעת  $members = 2$ , כלומר יש אדם מקבוצה 0 ואדם מקבוצה 1 באותו חדר באותו זמן, שזו שגיאת נכונות ומפר את כללי הכניסה לחדר.

דוגמת הרצה לתשובה c:

סטודנט מקבוצה 0 נכנס לחדר. כעת,  $members = 1 > 0$ . לאחר מכן סטודנט מקבוצה 1 נכנס לחדר מתחיל להפעיל את  $OnArrival(1)$ . סטודנט זה יבקש את המנעול בלולאה כל 10 שניות במקום לעבור להמתנה ובכך נעשה בדיקות מיותרות שיעלו לנו ב-10 שניות כל אחת (ובנוסף נבזבז זמן גם בהחלפות הקשר), מה שיגרום לבזבוז של זמן ולמעשה ל- $busywait$ .

דוגמת הרצה לתשובה d:

נניח כי מגיע אדם מקבוצה 1 ומתחיל להפעיל את  $OnArrival(1)$ . כעת  $members = 0$  ולכן המנעול יינעל, לולאת ה- $while$  לא תתבצע והמנעול ייפתח. ונבצע  $member++$ . כעת, יגיע אדם נוסף מקבוצה 1 וגם הוא מתחיל להפעיל את  $OnArrival(1)$ . כעת  $members = 1$  ולכן ניכנס ללולאה ונמתין עד שהאדם הראשון שנכנס יצא מהחדר. כלומר קיבלנו כי שני אנשים מאותה קבוצה לא הצליחו להיכנס לחדר באותו הזמן למרות שהם מאותה הקבוצה, ולכן והדבר מפר את כללי הכניסה לחדר.

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

נניח שיש 5 אנשים מקבוצה 1 בחדר, ונרצה להכניס אדם אחד מקבוצה 0, לכן נקרא ל- $OnArrival(0)$ . כעת,  $members = [0, 5]$ , ו- $team = 0$ ,  $other = 1$ , ולכן  $members[other] = 5 > 0$ , ולכן תנאי הלולאה יתקיים והחוט יעבור להמתנה עד שיתקיים התנאי  $cond[0]$  והמנעול ישתחרר. כעת נקרא ל- $OnLeave(1)$  כלומר נוציא אדם מקבוצה 1 מהחדר, כעת יינעל המנעול ויתבצע --  $members[1] = 4$  כלומר כעת  $members[1] = 4$  יישלח סיגנל אל  $cond[0]$  וייפתח המנעול, וכעת ימשיך הקוד בחוט של  $OnArrival(0)$  כי הוא קיבל את הסיגנל שציפה לו, והמנעול פנוי לכן ינעל אותו וימשיך בקוד. הוא יבדוק את תנאי לולאת ה- $while$  ומתקיים כי  $members[other] = 4 > 0$ , ולכן תנאי לולאת יתקיים והחוט יעבור שוב להמתנה עד שיתקיים התנאי  $cond[0]$  והמנעול ישתחרר. כלומר קוד זה גורם לחוט להתעורר ללא צורך מפני שהוא חוזר מיד לאחר מכן למצב המתנה. מצב זה ייקרה שוב ושוב עד שייתקבל  $members[other] = 0$ , מה שיגרום למספר חוטים להתעורר ללא צורך.

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

ניתן לתקן את הבעיה שהציג עומר בסעיף הקודם על ידי הוספת השורה  $if(members[team] == 0)$  לפני שורה 16. באופן זה, יישלח סיגנל לחברי קבוצה A רק במקרה שבו אין חברי קבוצה מ-B בחדר. לכן, לא ייוצר מצב שבו יישלח סיגנל לחוטים שלא צורך, כי סיגנל יישלח לחוט רק במידה שהחדר פנוי מאנשים, וחברי הקבוצה האחרת יכולים להיכנס לחדר ללא חשש. והחוט מייצג חבר קבוצה כך שחברי הקבוצה השנייה לא נמצאים בחדר כלומר חבר קבוצה שיכול להיכנס לחדר.

1. 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 condition race למשאב משותף.

b. יתכן מצב שחבר קבוצה כלשהי לא נכנס לחדר למרות כלל הכניסה שמתיר זאת, עקב condition race למשאב משותף.

c. קיימת בעיית Livelock / DeadLock בקוד.

d. יתכן מצב שחבר קבוצה כלשהי יחכה למרות כלל הכניסה שמתיר זאת, כאשר אין condition race למשאב משותף.

דוגמת הרצה לתשובה a:

נניח כי אדם מקבוצה 1 רוצה להיכנס לחדר, נקרא ל- $OnArrival(1)$ , וכעת לאחר הסיום שלה יש לנו אדם אחד מקבוצה 1 בחדר. מתקיים כרגע כי  $members[1] = 1$ , ונרצה להוציא את האדם מקבוצה 1, נקרא ל- $OnLeave(1)$  ונבצע החלפת הקשר לאחר שורה 15, כאשר  $tmp = 1$ . לאחר החלפת ההקשר תתחיל ריצה של אדם נוסף מקבוצה 1 שרוצה להיכנס לחדר, נקרא ל- $OnArrival(1)$ , וכעת לאחר הסיום שלה מתקיים כי  $members[1] = 2$ . כעת בעזרת החלפת הקשר נחזור לשורה 16 בפונקציה  $OnArrival(1)$  של האדם הראשון, ובשורה זו אנו מעדכנים את  $members[1]$  בעזרת  $temp = 1$  שערכו כבר אינו מעודכן, לכן יגרום לעדכון השגוי:  $members[1] = 0$ . כלומר מערך האנשים מראה שיש 0 אנשים מקבוצה 1, למרות שיש אדם אחד מקבוצה 1 בחדר. כעת, נוכל להכניס אדם מקבוצה 0 לחדר בעזרת  $OnArrival(0)$ , וקיבלנו כי יש בחדר באותו הזמן שני אנשים מקבוצות שונות, בעקבות  $race condition$  למשאב משותף.

דוגמת הרצה לתשובה b:

נניח כי אדם מקבוצה 1 רוצה להיכנס לחדר, לכן נקרא ל- $OnArrival(1)$ , ניתן לה להסתיים וכעת מתקיים  $members[1] = 1$ . נניח כי אדם נוסף מקבוצה 1 רוצה להיכנס לחדר, לכן נקרא ל- $OnArrival(1)$ , ונבצע החלפת הקשר לאחר שורה 9, כאשר  $temp = 1$ . כעת נקרא ל- $OnLeave(1)$ , עבור החוט הראשון, מתקיים  $temp = 1$  ולכן בסיום הפונקציה  $members[1] = 0$ . כעת, נבצע החלפת הקשר לחוט השני שימשיך את  $OnArrival(1)$ , כאשר  $temp = 1$ , לכן הוא יעדכן:  $members[1] = temp + 1 = 2$ . לאחר מכן נבצע  $OnLeave(1)$ , יתקיים כי  $temp = 2$ , ולכן בסוף הפונקציה יתעדכן  $members[1] = 1$ , למרות שאין יותר אנשים מקבוצה 1 בחדר. כעת, נבצע  $OnArrival(0)$  עבור אדם מקבוצה 0, ולכן תנאי הלולאה  $members[1] > 0$  מתקיים ונבצע  $cond\_wait$ . אבל מכיוון שאין יותר אנשים מקבוצה 1 בחדר, אדם 0 ממתינ ליציאה של אדם שכבר לא בחדר ולכן ימשיך לחכות לנצח. כלומר, סך הכל קיבלנו מצב שבו אדם 0 לא מצליח להיכנס לחדר, למרות שמותר לו לפי כללי הכניסה, בשל  $race condition$  למשאב משותף.

דוגמת הרצה לתשובה d:

ראשית, נקרא ל- $OnArrive(0)$ , המנעול  $m\_arrival$  פנוי ולכן נכניס אדם מקבוצה 0 לחדר, מערך האנשים  $members$  בחדר יהיה מאותחל ל-0, ולכן תנאי הלולאה לא יתקיים והקודד ימשיך הלאה, להוסיף 1 לתא 0 במערך האנשים בחדר, המנעול  $m\_arrival$  ישתחרר. כעת אנו במצב בו יש אדם אחד מקבוצה 0 בתוך החדר. נקרא ל- $OnArrive(1)$ , כלומר ננסה להכניס אדם מקבוצה 1 לחדר בזמן שיש בו אדם מקבוצה 0. כעת תא  $0 < 1 = members[0]$  ולכן תנאי הלולאה יתקיים והחוט ייכנס להמתנה לסיגנל עבור  $cond[1]$  וגם המנעול  $m\_arrival$  ישתחרר עד שיתקיים התנאי. כעת מחוט שלישי, נקרא ל- $OnArrive(1)$ , ובדיוק כמו בחוט השני, נעבור להמתנה בתור לסיגנל עבור  $cond[1]$  וגם המנעול  $m\_arrival$  ישתחרר עד שיתקיים התנאי. כעת נקרא ל- $OnLeave(0)$ , המנעול של  $m\_leave$  פנוי ולכן הוא ייכנס ויתחיל ויעדכן את התא של אנשים מקבוצה 0 בחדר להיות 0, וכעת יישלח סיגנל עבור  $cond[1]$  וישתחרר המנעול של  $m\_leave$ . ביצוע  $cond\_signal(&conds[1])$  יעיר חוט אחד בלבד שמצפה לסיגנל  $cond[1]$ , נניח בה"כ כי זה חוט 2. כעת המנעול  $m\_arrival$  פנוי, לכן הוא ינעל אותו, וכרגע מתקיים  $members[0] = 0$  ולכן תנאי הלולאה לא מתקיים ולכן נוסיף את האדם מקבוצה 1 שבחוט השני לחדר ונעדכן  $members[1] = 1$  ולאחר מכן נשחרר את המנעול. נשים לב כי ברגע זה, יש רק אדם אחד מקבוצה 1 בחדר, ולכן גם האדם בחוט השלישי יכול להיכנס מפני שהוא בקבוצה 1, אך אין אף אחד שיעיר אותו ולכן יישאר בהמתנה לסיגנל  $cond[1]$  ולא ייכנס לחדר.

