

תרגיל בית #3 - יבש-2

מטרת התרגיל היא :
להעמיק את הבנתכם בנושא מקביליות (concurrency).





שאלה 1.

נתונה פעולה אטומית $\text{CAS}(x, \text{old}, \text{new})$ - compare \& swap . הפקודה בודקת האם ערך המשתנה x שווה ל old . אם כן, הפעולה מחליפה את ערכו של x ל new ומחזירה true . אחרת, הפעולה משאירה את x ללא שינוי ומחזירה false .
 x יכול להיות משתנה משותף, ואילו old ו new לא יכולים להיות משותפים.

השתמשו בפקודה CAS על-מנת לממש בפסודו-קוד פונקציה $\text{register}(\text{num})$ השומרת ערך num 0 בתא פנוי (המכיל אפס) במערך משותף.

הנחיות לפיתרון:

- המערך המשותף (הגלובאלי) $\text{int list}[\text{MAX_REGISTER}]$ מאוחל לאפסים.
- עליכם להבטיח שהמערך list מתמלא באופן רציף, כלומר אסור שיהיה תא שמכיל אפס לפני תא שמכיל ערך חיובי.
- כל המספרים שהוכנסו עם register חייבים להופיע במערך.
- מספר תהליכים (תהליכונים) יכולים להריץ את register במקביל, אבל ניתן להניח שהפונקציה register לא נקראת יותר מ MAX_REGISTER פעמים בסך הכל.
- הפתרון יכול לבצע busy wait .

שאלה 2.

נתון צומת בצורת T. מכוניות המגיעות מימין או משמאל חייבות להמשיך ישר, והן יכולות לעשות זאת בו זמנית. מכוניות המגיעות מלמטה חייבות להמתין עד שהצומת יתפנה, לפני שהן חוצות אותו. השתמשו בסמפורים וכתבו שתי פונקציות: $\text{enter}(\text{direction})$ ו- $\text{leave}(\text{direction})$ לניהול הכניסות והיציאות מהצומת. כך שתהליכון הרוצה להיכנס לצומת קורא לפונקציה enter עם הכיוון ($\text{down or right or left}$) ממנו הוא רוצה להכנס, ותהליכון היוצא מהצומת קורא ל- leave עם הכיוון ($\text{down or right or left}$) ממנו הוא בא.

על הפתרון להיות מקבילי ככל האפשר ולמנוע תאונות.

שאלה 3.

בפאב הטכניון שלושה ברמנים וסוג קוקטייל יחיד בלבד - המורכב משלושה מרכיבים : וודקה, תפוזים ומטריה.
בכל ערב מגיעים מספר סטודנטים, מזמינים את הקוקטייל ועל הברמנים להכין אותו מכל שלושת המרכיבים.

לכל אחד מהברמנים תפקיד :

1. מוזג וודקה

2. מוסיף תפוזים

3. שם מטריה

תמיד יש כוס אחת בלבד בהכנה להגשה ועל מנת שטעם הקוקטייל לא ייהרס, הוודקה חייבת להימזג לפני שמוסיפים תפוזים (את המטריה ניתן להוסיף בכל שלב בהכנת הקוקטייל). ניתן להניח כי פעולת ההגשה עצמה מתבצעת מיידית.

הניחו כי קיים המשתנה המשותף g מטיפוס glass אשר מסמן את הכוס שבהכנה :

glass g;

טיפוס glass תומך בשתי פונקציות :

• add(ingredient)

פונקציית הוספת מרכיב לכוס. ingredient יכול להיות vodka או orange או

umbrella.

• serve_if_ready()

פונקציה אטומית, המבצעת את סדר הפעולות הבא באופן אטומי :

אם (כל שלושת המרכיבים נמצאים ב-g) :

• הגש את g לסטודנט

• כוס חדשה g =

אחרת : <לא קורה כלום והתוכנית ממשיכה כרגיל>

* ל-add() ול-serve_if_ready() אין ערכי החזרה

* add() אינה אטומית

* על כל כוס שמוגשת להכיל בדיוק מרכיב אחד מכל סוג

א. ממשו פסאודו-קוד של הפונקציות עבור כל אחד מהברמנים (הניחו כי כל פונקציה נקראת בלולאה אינסופית) :

put_vodka()

put_orange()

put_umbrella ()

ב. ממשו פונקציית אתחול הנקראת בדיוק פעם אחת לפני שכל הברמנים מתחילים לעבוד :

init()

על הפתרון להיות נכון ולהיות מקבילי ככל האפשר (רק פתרון בעל מקביליות מקסימלית יקבל את מלוא הניקוד)!

שאלה 4.

המערך `students[N]` מתאר, עבור כל כיתה בביה"ס (מ 0 עד $N-1$), את מספר הבנים, מספר הבנות, ומספר התלמידים הכולל בכיתה:

```
struct class_nums {
    int girls;
    int boys;
    int total;
}
```

```
class_nums students[N];
```

הפונקציה `update(n, girls, boys)` מעדכנת את התא `n` במערך כדלקמן:

```
void update( int n, int girls, int boys) {
    students[n].girls = girls;
    students[n].boys = boys;
    students[n].total = girls + boys;
}
```

הפונקציה `get_all(n)` מחזירה את המידע לגבי מספר הילדים בכיתה:

```
class_nums get( int n) {
    return students[n];
}
```

1. המימוש אינו `thread-safe`, כלומר בהרצתו ע"י מספר תהליכונים (`threads`) בו זמנית, יתכן שהפעולות לא יתבצעו באופן אטומי.
הצג/י תרחיש מפורט המדגים את הבעיה.

2. שנו את מימוש הפונקציות לעיל כך שהבעיה תיפתר. חשוב לאפשר מקבילות רבה ככל האפשר.

- השתמשו במשתנים מטיפוס `semaphore` התומכים בפעולות `up(semaphore s)` ו `down(semaphore s)`.
- יש להגדיר את המשתנים בהם משתמשים.

3. הוסיפו פונקציה חדשה, `move_girl(int n1, int n2)`, המעבירה בת אחת מכיתה `n1` לכיתה `n2`. ניתן להניח כי `n1` שונה מ-`n2`.

בהצלחה!!!