

תרגיל בית 2

אופן ההגשה:

- תשובות בכתב הקפידו להזין לתוך קובץ pdf תוך ציון מספר השאלה ומספר הסעיף. אפשר לכתוב ביד ולסרוק.
- ענו בתמציתיות ודיוק על השאלות בכתב.
- יש לרשום את **שם ות"ז** המגישים בראש קובץ pdf וקבצי הקוד.
- את כלל קבצי העבודה יש לכווץ בפורמט zip ולהגיש קובץ זה.
- יש להגיש את הפתרון עד למועד שנקבע בלימוד.
- שאלות יש לשאול בפורום.

בהצלחה!

שאלה 1 – שאלה תכנותית (בשאלה זו יש להגיש קצבי java) (60 נק')

ברפת מסוימת קיימים 3 מתחמים: שוקת, חציר, ואזור פתוח לטיול.

ברפת ישנן N פרות, אשר עליהן לציית לכללים הבאים:

- בכל זמן נתון, יהיו עד X פרות במתחם החציר
- בכל זמן נתון, יהיו עד Y פרות במתחם השוקת
- פרה יכולה לצאת לטיול, רק לאחר שכל האחרות סיימו במתחם החציר והשוקת
- פרה יכולה לגשת לשוקת רק לאחר שסיימה בחציר
- כל פרה נמצאת במתחם כלשהו בין 2 ל-10 שניות (יש להגריל את המספר לכל פרה ולכל מתחם)

כל הפרות מעוניינות לעבור בכל אחד מהמתחמים פעם אחת בדיוק

א. ע"י שימוש ב semaphores, כתבו אלגוריתם עבור תהליך/תהליכון הנקרא Cow המעביר את הפרה בכל המתחמים פעם אחת בדיוק תחת הכללים שהוזכרו. השתמשו במתודות הבאות כדי להודיע על כניסה למתחם מסוים:

drink() – הפרה במתחם השוקת

eat() – הפרה במתחם החציר

walk() – הפרה במתחם הטיול

שים לב שהאלגוריתם שנכתב עבור Cow זהה עבור כל N התהליכים של הפרות ברפת.

ב. באמצעות semaphore, ממשו את האלגוריתם מסעיף א כתוכנית ע"פ ההנחיות הבאות:

1. כתבו מחלקה בשם Compounds המכילה פונקציות walk, eat, drink אשר ידפסו הודעה באיזה מתחם נמצאת הפרה. על כל פונקציה גם לבדוק שפרה לא מפרה את חוקי הרפת. למשל, פונקציה drink תבדוק שפרה הייתה כבר בחציר, שיש לא יותר מ-Y פרות בשוקת אחרי שהיא נכנסה לשוקת וכו'. במידה ופרה מפרה את חוקי הרפת יש להדפיס הודעה מתאימה.
2. צרו מחלקה בשם SharedResources ובתוכה הגדירו את כלל ה semaphores שנדרשים למימוש האלגוריתם.
3. צרו מחלקה CowThread, המהווה תהליכון בתוכנית, ולתוכה הזינו את מימוש האלגוריתם מסעיף א. במחלקה קיימים:
 - a. משתנה final int id, המשמש id ייחודי של הפרה, משתנה comp מסוג Compounds ומשתנה res מסוג SharedResources.
 - b. המחלקה מקבלת בבנאי id ייחודי של הפרה, אובייקט SharedResources (דרכו משתמשת ב semaphores שהגדרתם – למשל (res.sem1.up()), ואובייקט Compounds שדרכו קוראת למתודות walk(), eat(), drink() עם המזהה של הפרה (לדוגמא: comp.drink(id))
 4. צרו מחלקה נוספת שתשמש כ main בתוכנית. בתוכה:
 - a. צרו אובייקט SharedResources i Compound (רק אחד מכל אחד)
 - b. בקשו מהמשתמש לבחור את סה"כ כמות הפרות N וגם כן ערכים של X ו-Y.
 - c. צרו N תהליכונים והריצו אותם (שלחו אליהם בבנאי את האובייקטים שיצרתם)
 5. צרו מחלקות של כל המתחמים. בכל מתחם יש לשמור רשימה של הפרות אשר נמצאות ברגע נתון במתחם. כאשר פרה נכנסת למתחם יש לשמור אובייקט של הפרה ברשימה ומיד אחרי שפרה מסיימת במתחם יש למחוק אותה מרשימת הפרות במתחם.
 6. מותר להוסיף שדות ומטודות למחלקות שתוארו וגם כן להוסיף מחלקות עזר נוספות.

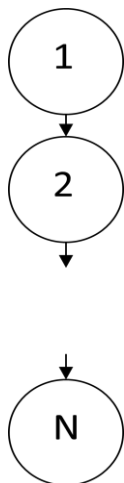
ג. מדוע דרוש לנו האובייקט SharedResources? למה לא ניתן פשוט ליצור את ה semaphores בתוך המחלקה CowThread?

שאלה 2 (20 נק')

נתון הקוד הבא, המבצע שימוש בפונקציה fork :

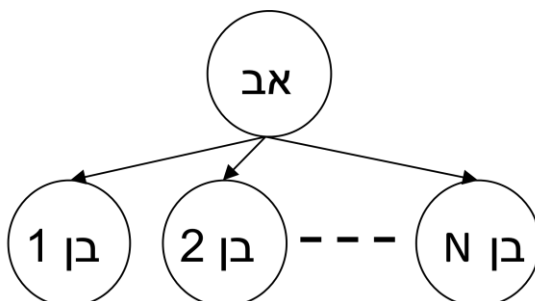
```
void main(){
    int i=0;
    while(i<5){
        pid = fork();
        if (pid == 0)
            printf("hello");
        else
            printf("world");
        i++;
    }
}
```

- כמה פעמים תודפס המילה "hello" וכמה פעמים תודפס מילה "world"?
- ציירו את היררכיה של התהליכים שנוצרו.
- כתובו תוכנית (בקובץ pdf) אשר תקבל ממשתמש מספר שלם חיובי N ותיצור שרשרת תהליכים:



יש להדפיס pid של כל אחד מהתהליכים שנוצרו.

- כתוב תוכנית אשר תקבל מהמשתמש מספר שלם חיובי N ותיצור עץ תהליכים בגובה 1 (תהליך אב ו-N בנים שלו):



יש להדפיס pid של כל אחד מהתהליכים שנוצרו.

שאלה 3 (10 נק')

נתון קטע הקוד:

```
int main(int argc, char *argv[]){  
  
    for(int i = 0; i < 7; i++)    {  
        int pid = fork();  
        printf("%d : %d , \n", getpid(), i);  
        if(pid > 0)  
            break;  
    }  
    return 0;  
}
```

הניחו כי ה-pid של התהליך שמתחיל להריץ את ה-main הוא 1, וכי pids ניתנים באופן סדרתי, ואין עוד תהליכים במערכת. כמו כן, הניחו שיטת התזמון First Come First Served. כלומר, תהליך שנכנס לתור ראשון ירוץ ראשון.

- א. מה ה-pid הגבוה ביותר שהוקצה במהלך הקוד להלן?
ב. מה ידפיס הקוד (משמאל לימין)?

שאלה 4 (5 נק')

להלן תבנית של תוכנית עם שני תהליכים ופונקציית main.

main()	p1()	p2()
	{start(); while (TRUE){ server_create(); finish();} }	{ while (TRUE) {before(); client_use(); cleanup();}}

התניות:

- א. קטע start צריך לסיים לרוץ לפני שקטע before ירוץ אפילו פעם אחת.
ב. קטע server_create הוא יצרן וקטע client_use הוא הצרכן של מה שהיצרן יצר.
ג. קטעים before, finish, cleanup הם קטעים קריטיים אחד לשני ואסור שירוצו יחד.

הוסף לפסאודו-קוד הנ"ל פקודות של סמפור כדי לקיים את שלוש ההתניות. השתמש אך ורק בפקודות הבאות: wait(s), signal(s), init_sem(s,n).

שאלה 5 (5 נק')

נתונים 3 תהליכים שרצים במקביל:

main	P1	P2	P3
	{ B E }	{ A C }	{ D F }

התניות:

- קטע A חייב להסתיים לפני שמתחילים B, C ו-D.
- קטעים B, C, D הם קטעי קוד קריטי ואסור שיפעלו באותו זמן.
- קטע E תתבצע רק אחרי סיום B, C ו-D.

תעתיקו את התהליכים לקובץ בתוספת מנגנון שמקיים את ההתניות בעזרת סמפורים.