**מעבדה במערכות הפעלה: מפגש 10**

**נושא: שימוש בחוטים בסביבת LINUX**

**הגשה:**  הגשה אחת עבור מפגשים 10-11

**משימה 1: לסנכרן הדפסות בין חוטים**

* התוכנית **threads\_char\_by\_char.c** מייצרת מספר חוטים, המריצים אותו קוד, שמדפיסי מחרוזת.
* תקמפלו ותריצו את התוכנית ותראו אם אתם רואים בעיה בהדפסות
  1. הערה: לקמפל תוכנית שמשתמשת בחוטים צריך להוסיף **lpthread-** בסוף פקודת קומפילציה
* תפתרו את הבעיה שרואים עיי' כך שתאפשרו גישה למסך (קטע קריטי) רק לחוט אחד בו זמנית.
* **רמז:** תשתמשו בטיפוס **pthread\_mutex\_t** להגדיר משתנה שישמש לנעילה (תחפשו באינטרנט איך להשתמש בטיפוס זה להגדרת מנעול ושימושו).

הפעולות הדרושות הן:

* + - * אתחול המשתנה
      * נעילה / שחרור

**הגשה (בקובץ אחד):**

* תוכנית מתוקנת
* הדפסת התוכנית המקורית
* הדפסת התוכנית המתוקנת

**משימה 2: עדכון מונה משותף מתוך מספר חוטים**

* התוכנית **thread\_counters.c**  מייצרת שני חוטים המריצים אותו קוד המעדכן מונה משותף. בסיום שני החוטים התוכנית מדפיסה את ערך הסופי של המונה.
* מה הערך הצפוי של המונה בסיום התוכנית ? מה תוצאת ההרצה ?
* תשנו את התוכנית כדי שתוצאה תהיה כמו מצופה.

**הגשה (בקובץ אחד):**

* תוכנית מתוקנת
* הדפסת התוכנית המקורית
* הדפסת התוכנית המתוקנת

**משימה רשות (המשך של משימה 2):**

* במשימה 2 אפשר לעשות נעילה בשני מקומות:
  1. מיד תחילת החוט
  2. עבור כל פעולת עדכון המונה

**מה ההבדל ?**

**הגשה:**

* תריצו את שני הפתרונות עבור ערכים הבאים של MAX\_COUNTER ותמלאו את הטבלה ( תציינו בערך מספר שניות):

|  |  |  |
| --- | --- | --- |
| **זמן עבור נעילה מסוג 2** | **זמן עבור נעילה מסוג 1** | **MAX\_COUNTER** |
|  |  | **1.000.000** |
|  |  | **20.000.000** |
|  |  | **100.000.000** |
|  |  | **200.000.000** |

* תסבירו במשפט אחד למה בשיטה 2 הזמן יותר ארוך
* תסבירו בקצרה מה יתרון של כל אחת משיטות