

תרגיל מס 2 - UNIX System Programming

הנחיות בסיסיות:

- (1) ניתן להגיש בזוגות.
- (2) תרגילים אלו מהווים חלק ניכר בציון הסופי של הקורס. לכן, העתקות יענו בחומרה רבה. עדיף להגיש תרגיל קצת פחות טוב, אבל שעבדתם עליו לבד/בזוג, מאשר להגיש תרגיל מועתק.

הנחיות הגשה:

- (1) יש לכלול את כלל קבצי ה source (קבצי ה c) וה headers (קבצי ה h) בלבד
- (2) התרגיל ייבדק ב Linux. על כן, לסטודנטים שעובדים במערכות אחרות, אני ממליץ לבדוק הקוד גם ב Linux
- (3) במידה ועליי לדעת דברים נוספים אודות פתרוןכם, צרפו קובץ README המכיל דברים אלה.

תרגיל 1 - חלק א':

כתבו מנהל בריכת פתילים בעל מספר פתילים קבוע. על מנהל בריכת הפתילים לספק API המכיל לפחות את הפונקציות הבאות:

```
int ThreadPoolInit(struct ThreadPoolManager* t, int n);
```

בהינתן מנהל בריכת פתילים (לא מאותחל), ומספר פתילים נדרש n , מאתחל בריכת פתילים בעלת n פתילים בדיוק. על הפונקציה להחזיר 0 במקרה של הצלחה וערך שלילי כלשהו במקרה של כשלון.

```
int ThreadPoolInsertTask(struct ThreadPoolManager* t, struct Task* task);
```

בהינתן מנהל בריכת פתילים ומשימה, על מנהל בריכת הפתילים להכניסה לתור המשימות. משימה מוגדרת על פי המבנה הבא:

```
struct Task
{
    void* (*f)(void*);
    void* arg;
};
```

על הפונקציה להחזיר 0 במקרה של הצלחה וערך שלילי כלשהו במקרה של כשלון.

```
void ThreadPoolDestroy(struct ThreadPoolManager* t);
```

בהינתן מנהל בריכת פתילים, משחרר את המשאבים בהם משתמשת בריכת הפתילים (למשל, תור המשימות והפתילים עצמם).

השתמשו ב *mutex* ו *condition variable* לסנכרון הגישה לתור והתראה על משימה חדשה.
השתמשו בתור המכיל איברים מסוג *struct Task*. ניתן להשתמש ב *std::queue* ב *STL* או לחילופין לממש תור מבוסס רשימה מקושרת ב *C*.