

## מעבדה 9 - סמפורים – בעיות נוספות

To initialize a semaphore, use [sem\\_init](#):

```
int sem\_init(sem_t *sem, int pshared, unsigned int value);
```

- **sem** points to a semaphore object to initialize
- **pshared** is a flag indicating whether or not the semaphore should be shared with fork()ed processes. In Linux only 0;
- **value** is an initial value to set the semaphore to

Example of use:      `sem_init(&sem_name, 0, 10);`

---

To wait on a semaphore, use [sem\\_wait](#):

```
int sem\_wait(sem_t *sem);
```

Example of use:      `sem_wait(&sem_name);`

- If the value of the semaphore isn't positive, the calling process blocks; one of the blocked processes wakes up when another process calls `sem_post`.
- 

To increment the value of a semaphore, use [sem\\_post](#):

```
int sem\_post(sem_t *sem);
```

Example of use:

```
sem_post(&sem_name);
```

- It increments the value of the semaphore and wakes up a blocked process waiting on the semaphore, if any.
- 

## תרגיל 1.

עליכם לממש קוד למחסנית משותפת ל K חוטים.

כלומר, עליכם להגדיר מערך גלובלי: `int stk[N]` לייצוג המחסנית ואינדקס `int idx` ועליכם לכתוב את הפונקציות:

```
void stkPush(int num);
```

```
int stkPop();
```

שימו לב: חוט המבצע POP ממחסנית ריקה, יחכה שיהיה מה להוציא.

חוט המבצע PUSH למחסנית מלאה, יחכה עד שיתבצע POP ויהיה מקום

ה main יהיה:

```
#define K 5

#define N 10

void stkPush(int num);

int stkPop();

...//ליתן להוסיף קוד כרצונך...

int main(int argc, char * argv[]){

    pthread_t tArr[K];

    int i=0 , ans[K];

    ...//ליתן להוסיף קוד כרצונך...

    for(i=0; i<K; i++)

        ans[i]=pthread_create(&tArr[i], NULL, tFunc, NULL);

    sleep(10);

    return 0;

}

void* tFunc(void* p) {

    int num;

    while(1){

        num = rand()%100;

        if(rand()%2) {

            stkPush(num);

            printf("Push %d\n", num);

        } else {

            printf("Pop %d\n", stkPop());

        }

        sleep(1);

    }

}
```

## תרגיל 2

לטובת הבידוד, בחנות "בריאותה" הפרידו בין אנשים מבוגרים (מעל 70) וצעירים (מתחת ל 70).

אם בחנות נמצא כרגע מבוגר (70 ומעלה), לא יוכל להיכנס לחנות איש נוסף עד שהמבוגר יסיים את הקניות בחנות.

אם בחנות נמצא כרגע צעיר, יכולים להיכנס לחנות עוד צעירים נוספים אחרים ולא מבוגרים.

כל איש הרוצה לקנות בחנות, מקבל מספר סידורי תחילה.

עליך להגדיר את הפונקציות הבאות:

- void oldManWantsToBuy(void\* ind) - עבור קונה מעל 70 בעל מספר סידורי ind שרוצה לקנות בחנות.

- void youngManWantsToBuy(void\* ind) - עבור קונה צעיר בעל מספר סידורי ind שרוצה לקנות בחנות.

בנוסף, יש לכתוב פונקצית main היוצרת N חוטים שידמו מבוגרים ו M חוטים שידמו צעירים שיפעילו את הפונקציות הנ"ל. כל חוט המדמה מבוגר יפעיל את הפונקציה oldManWantsToBuy וכל חוט המדמה צעיר יפעיל את הפונקציה youngManWantsToBuy, אשר מטרתן לדמות את הגעת הקונה לחנות, קנייה ועזיבת החנות.

נדרש בין היתר להדפיס את ההודעות הרלוונטיות (דוגמא לפלט בהמשך).

כתשובה לשאלה זאת יש לכתוב תכנית מתועדות בשפת C.

חובה להשתמש בסמפורים. נדרש להוסיף קוד לאתחול הסמפורים ב main.

M, N יוגדרו כרצונכם. על התוכנית להיות נכונה לכל M, N.

לדוגמא, פלט אפשרי עבור 2 מבוגרים (N=2) ו 3 צעירים (M=3):

```
Old man #0 wants to buy
Old man #0 is buying
Old man #1 wants to buy
Old man #0 is done
Old man #1 is buying
Young man #0 wants to buy
Young man #1 wants to buy
Young man #2 wants to buy
Old man #1 is done
Young man #0 is buying
Young man #1 is buying
Young man #2 is buying
Young man #1 is done
Young man #2 is done
Young man #0 is done
```