

מבוא למערכות הפעלה – תרגיל 1

מרצה: דר' כרמי מרימוביץ

שימו לב: תרגיל זה דורש לקרוא תיעוד של Win32.

1. בכניסה לבניין מותקנת דלת מסתובבת. מטרתכם לכתוב תוכנית שתאפשר כניסה ויציאה מהבניין ביעילות, ללא פגיעה בנכנסים ויוצאים, ותחסוך בחשמל. לרשותכם שגרה rotate90 שמסובבת את הדלת ב-90 מעלות (הדלת מסתובבת בכיוון אחד בלבד). כיון שאין לנו באמת דלת על שגרה זו רק להדפיס שהדלת מסתובבת. אין ידיעה או שליטה על זמן הסיבוב. להלן הפרטים הטכניים: לדלת יש ארבעה תאים, בכל תא יש מקום לכל היותר לאדם אחד. תא אחד פונה אל מחוץ לבניין, תא אחד פונה אל פנים הבניין, ושני תאים חסומים. ישנם שני תורים, מחוץ לבניין (כדי להיכנס) ובתוך הבניין (כדי לצאת). כאשר יש תא ריק, מי שבראש התור נכנס לתוכו. אם בשני התורים יש אנשים, ובהתאמה יש תאים פנויים, צריך למלא את שניהם. סיבוב הדלת תוך כדי כניסה או יציאה משמעו פגיעה בניכנס או יוצא. הסדר חשוב: אם מגיע תא מלא, קודם יוצאים ממנו ורק אז נכנסים אליו.

2. עליכם לכתוב **תוכנית** שתממש את הנ"ל. אדם שרוצה להיכנס/לצאת יהווה חוט. עליכם לספק בין השאר את השגרות BuildingEnter ו-BuildingLeave להן יקראו החוטים. שגרות אלו חוזרות לקורא לאחר שהאדם נכנס/יצא מהבניין בביטחה. לרשותכם סמפורים, מוטקסים וכיו"ב. ניתן להניח שהתעוררות מסמפור/מוטקס תקוע היא על-ידי תור fifo.

3. על התוכנית הראשית לחכות פרק זמן אקראי של עד משניה ולייצר חוט ולחזור על כך. הארגומנט הראשון לתוכנית יהיה מספר החוטים המקסימלי המותר. באם הגענו למספר זה, יש לחכות עד שחוט יסיים ואז להמשיך. כל חוט "יכנס לבניין" יחכה שם פרק זמן אקראי של עד שתי שניות, "יצא מהבניין", ויסיים. על כל חוט להדפיס את מצבו.

4. משתמשי VC++ 6.x שימו לב. ספריות ברירת המחדל אינן מתפקדות טוב (בפרט I/O) בסביבה רבת חוטים ויש להחליפן בספריות המתאימות לסביבה זו.

פונקציות API שמושיות:

1. Win32 threads:

```
#include <windows.h>
DWORD tid;
HANDLE thandle;
```

```
thandle = CreateThread(
    NULL,           // Default security attribute
    0,              // Default stack size
    func,           // function to run
    arg,            // argument
    0,              // default creation flag
    &tid            // thread id
);
```

```
DWORD WINAPI func(DWORD arg)           // thread function
```

2. Win32 mutex:

```
#include <windows.h>
HANDLE mutex;

Mutex = CreateMutex(NULL, FALSE, NULL);
WaitForSingleObject(mutex, INFINITE);
// ** Critical section **
ReleaseMutex(mutex);
```

3. Win32 semaphore:

```
#include <windows.h>
HANDLE sem;

Sem = CreateSemaphore(NULL, 0, 5, NULL);
WaitForSingleObject(sem, INFINITE); // Acquire semaphore
// ** relevant code **
ReleaseSemaphore(sem, 1, NULL); // Release semaphore
```

4. שימו לב, כדאי מאוד (מאוד) לקרוא את התיעוד של הפונקציות הנ"ל ולא להשתמש בהן באופן לא מושכל.

אין לעבוד בקבוצות

התיעוד משמעותי לציון.

שפות רלוונטיות: C/C++ ללא MFC/ATL וכיו"ב.

את התרגילים יש להגיש לתיבת התרגילים של הקורס.