1. מנגנוני אסינכרון של מאורעות כדי שיוצגו במקביל או בזמנים שונים
2. Manual reset event - "שער ידני" האפשרות לתריידים לפעול נחסמת ע"י שער שנפתח ונסגר באמצעות פקודה
3. Auto rest event - "שער אוטומטי" השער נסגר אוטומטית אחרי כל תרייד שנכנס
4. ההבדל בין המנגנון הזה ל - Wait Pulse הוא שכאן אין critical section ואין צורך במפתח כי התריידים לא יכולים להתנגש, הצורך הוא בארגון סדר הפעולות
5. שער נסגר ע"י פקודה RESET או שמלכתחילה הוא במצב סגור
6. שער נפתח ע"י פקודה SET או שמלכתחילה הוא במצב פתוח

mre.Reset();

private static ManualResetEvent mre = new ManualResetEvent(False);

mre.Set();

private static ManualResetEvent mre = new ManualResetEvent(True);

1. בפקודה mre.WaitOne(5000); יש להגדיר זמן למשל 5 שניות (5000 אלפיות שניה)
2. Task – יחידת עבודה שיכולה לנהל עבודה אסינכרונית, המכילה בתוכה הליכונים(תריידים) בעלי תפקידים שונים לביצוע המשימה

Thread – הליך עם תפקיד מסוים שעליו לבצע

יתרונות:

1. ל Task תכונות שאין ל Thread
2. ה Task מושך Threads ישירות מה Thread Pool
3. Taskd יכולה להחזיר תוצאה
4. הכתיבה קלה יותר ב Task
5. Wait- המתנה רגילה

RunSynchronously(); - Start ו Wait ביחד

static void Main(string[] args)

{

Task<int> t = new Task<int>(() =>

{

return 7;

});

t.Start();

Console.WriteLine(t.Result);

}

1. Task.WaitAll(t1,t2) – המתנה לסיום כל הפעולות בסוגריים

Task.WaitAny(t1,t2) – המתנה שאחת מהפעולות בסוגריים תסתיים

1. Task Parallel Library (TPL) – ספריית משימות מקבילות,

ה TPL מארגן בעצמו את המשימות, מוסיף משימות מקבילות וממקסם את הביצועים של הקוד