

שאלה 2

א.

מדידה	Process (ms)	Threads(ms)
1	140.3954	15.1295
2	76.7258	12.3327
3	35.422	6.5309
4	57.971	5.4631
5	39.0203	7.7375
6	35.1813	26.1227
7	50.029	7.4585
8	52.9909	12.4097
9	62.4711	12.0845
10	62.1431	6.4683

ב. הסבר לתוצאות הטבלה :

בתרגיל מדדנו את זמן יצירת והרצת הודעת "Hello World" גם באמצעות יצירת תהליך חדש (Process) וגם באמצעות ת'רד 10 פעמים כל אחד.

התוצאות הראו שזמן יצירת תהליך היה גבוה משמעותית (בטווח של כ-35 עד 140 מילישניות), בעוד שזמן יצירת ת'רד היה נמוך בהרבה (כ-5 עד 26 מילישניות).

הסיבה לכך היא שבתהליך חדש, מערכת ההפעלה צריכה להקצות זיכרון נפרד, לטעון קבצים ולפתוח משאבים מחדש — תהליך כבד.

ת.ז 208730861

ת.ז 206850596

לעומת זאת, יצירת ת'רד קלה ומהירה הרבה יותר, כי הת'רד משתמש בזיכרון ומשאבים שכבר קיימים בתוך התהליך הראשי.

הסיבה היא כי תהליך מנוהל בעזרת PCB מבנה נתונים גדול שמכיל מידע כמו זיכרון, קבצים פתוחים, מצב ריצה ועוד.

לעומת זאת, כל ת'רד מנוהל בעזרת TCB מבנה נתונים קטן יותר שמכיל בעיקר מידע על רישום מצב הת'רד, מזהה, סטאק ועוד.

לכן גם מבחינת ניהול פנימי, תהליכים דורשים יותר משאבי מערכת מאשר ת'רדים.

תרגיל 3:

עבור 20 טרד'ס:

```
0 references
public static void Main(string[] args)
{
    Console.WriteLine($"Process ID: {Process.GetCurrentProcess().Id}");
    int rowsA = 1000;
    int colsA = 5000;
    int colsB = 1000;
    int numThreads = 20;
    // Initialize matrices
}
```

קיבלנו 38 טרד'ס ב task manager.

WD Discovery.exe	28308	Running	U2	01	19,972 K	30	x86	WD Discovery
msedge.exe	28248	Running	U2	00	52 K	7	x64	Microsoft Edge
GoogleDriveFS.exe	28224	Running	U2	00	648 K	8	x64	Google Drive
msedgewebview2.exe	28192	Running	U2	00	692 K	1	x64	Microsoft Edge WebView2
Widgets.exe	28096	Running	U2	00	6,604 K	22	x64	Widgets.exe
OP2.exe	28088	Running	U2	80	50,616 K	38	x64	OP2
conhost.exe	28068	Running	U2	00	124 K	2	x64	Console Window Host
GoogleDriveFS.exe	28044	Running	U2	00	520 K	12	x64	Google Drive
msedgewebview2.exe	27984	Running	U2	00	256 K	21	x64	WebView2 Utility: Network Service
AdobeCollabSync.exe	27948	Running	U2	00	580 K	8	x64	Acrobat Collaboration Synchronizer 25.1
MSBuild.exe	27596	Running	U2	00	26,996 K	15	x64	MSBuild.exe
Microsoft.ServiceHu...	27520	Running	U2	00	9,244 K	14	x64	Microsoft.ServiceHub.Controller
svchost.exe	27440	Running	U2	00	352 K	3	x64	Host Process for Windows Services
msedgewebview2.exe	27428	Running	U2	00	68 K	7	x64	Crashpad
conhost.exe	27276	Running	U2	00	96 K	2	x64	Console Window Host

נסיק כי יש עוד טרד'ס שלא קשורים לקוד עצמו אבל מופעלים דרך התוכנית כברירת מחדל.

ת.ז 208730861
ת.ז 206850596

דוגמה נוספת,

עבור 2 טרד'ס:

```
//main
0 references
public static void Main(string[] args)
{
    Console.WriteLine($"Process ID: {Process.GetCurrentProcess().Id}");
    int rowsA = 1000;
    int colsA = 5000;
    int colsB = 1000;
    int numThreads = 2;
    // Initialize matrices
    int[,] matrixA = new int[rowsA, colsA];
    int[,] matrixB = new int[colsA, colsB];
    int[,] resultMatrix = new int[rowsA, colsB];
    // Fill matrixA with random values
```

קיבלנו 20 טרד'ס ב task manager.

Details								
Name	PID	Status	User name	CPU	Memory (a...	Threads	Archite...	Des
SystemSettings.exe	11564	Suspended	U2	00	0 K	35	x64	Set
jusched.exe	11680	Running	U2	00	28 K	1	x86	Jav
SoftwareUpdateNoti...	11972	Running	U2	00	896 K	3	x86	Soft
chrome.exe	12104	Running	U2	00	18,812 K	21	x64	Go
conhost.exe	12208	Running	U2	00	368 K	4	x64	Co
WDDiscoveryMonito...	12256	Running	U2	00	644 K	5	x86	WD
conhost.exe	12472	Running	SYSTEM	00	216 K	3	x64	Co
OP2.exe	12632	Running	U2	16	47,768 K	20*	x64	OP
OneDrive.exe	12648	Running	U2	00	6,804 K	27	x64	Mic
conhost.exe	12688	Running	U2	00	60 K	2	x64	Co
backgroundTaskHos...	12724	Suspended	U2	00	0 K	13	x64	Back
WD Discovery.exe	12736	Running	U2	00	20 K	2	x86	WD
SecurityHealthServic...	12860	Running	SYSTEM	00	2,048 K	11		Win
chrome.exe	13006	Running	U2	00	62,276 K	10	x64	Go

שאלה 4:

א.

הסבר אלגוריתם:

האלגוריתם MTMergeSort הוא מימוש של אלגוריתם מיון מיזוג (Merge Sort) עם תמיכה בריבוי תהליכים (Multi-threading).

1. חלוקה:

· האלגוריתם מחלק את המערך לשני חלקים שווים או כמעט שווים עד שמגיעים למערכים בגודל של אלמנט אחד.

2. ריבוי תהליכים:

- אם גודל המערך הנוכחי גדול מסף מסוים ($nMin$) האלגוריתם יוצר שני תהליכים נפרדים (Threads) כדי למיין את שני החלקים במקביל.
- אם גודל המערך קטן או שווה ל $nMin$ -המיון מתבצע בצורה סדרתית.

3. מיזוג:

- לאחר שכל חלק מיון, האלגוריתם ממזג את שני החלקים הממוינים למערך אחד ממויין.
- המיזוג מתבצע על ידי השוואת האלמנטים בשני החלקים והעתקתם למערך זמני בסדר עולה.

4. העתקה חזרה:

- האלמנטים הממוינים מועתקים חזרה למערך המקורי.

האלגוריתם

1.0 פונקציית מיון_מיזוג_מרובה_תהליכים(רשימה, nMin) :

1.1 אם אורך(רשימה) $= 1$:

החזר רשימה

1.2 חצה את הרשימה לשתי תתי-רשימות:

1.2.1 שמאלית = החצי הראשון

1.2.2 ימנית = החצי השני

1.3 אם אורך(רשימה) $< nMin$:

1.3.1 קרא מיון_מיזוג_מרובה_תהליכים(שמאלית, nMin)(תהליך נפרד)

1.3.2 קרא מיון_מיזוג_מרובה_תהליכים(ימנית, nMin)(תהליך נפרד)

1.3.3 התחל את שני התהליכים

1.3.4 המתן לסיום שני התהליכים (Join)

אחרת

1.4 החזר פונקציית-מיזוג(שמאלית, ימנית)

סוף פונקציה(1)

2.0 פונקציה מיזוג(שמאלית, ימנית):

2.1 צור רשימה ריקה result.

2.2 כל עוד יש אלמנטים בשמאלית ויש אלמנטים בימנית :

אם השמאלי קטן או שווה לימני (באמצעות השוואת string.Compare)

2.2.1 הוסף את האלמנט השמאלי ל-result

2.2.2 הסר את האלמנט מהשמאלית

אחרת

2.2.3 הוסף את האלמנט הימני ל-result

2.2.4 הסר את האלמנט מהימני

2.3 הוסף ל-result את כל האלמנטים הנותרים בשמאלית

2.4 הוסף ל-result את כל האלמנטים הנותרים בימנית

2.5 החזר result

סוף פונקציה(2)

2.

