מבוא לתיכנות מונחה עצמים – תיכנות מקבילי

תיכנות מקבילי – למה צריך את זה?

1) **תגובתיות –** המחשב מבצע פעולה שלוקחת הרבה זמן. אנחנו רוצים שימשיך לבצע פעולות אחרות במקביל.

2) **הגבלת זמן –** אנחנו מבצעים חישוב שיכול להימשך הרבה זמן, אנחנו רוצים לעצור אותו כשנגמר לנו הזמן.

3) **מהירות –** אנחנו רוצים לנצל את כל הליבות של המחשב שלנו כדי לבצע חישוב כבד במהירות.

בעיה לדוגמה

בעיית החלוקה:

נתונה רשימה של מספרים.

צריך לחלק אותה לשתי תת-רשימות,

כך שההפרש בין שני הסכומים הוא קטן ביותר.

(יישום: חלוקה הוגנת של תכשיטים בין יורשים).

לא ידוע אלגוריתם הפותר את הבעיה בזמן פולינומיאלי.

שאלה: איזה אלגוריתם פשוט פותר את הבעיה, ובכמה זמן?

תיכנות מקבילי בבעיית החלוקה

1) **תגובתיות –** שרת-רשת לחישוב חלוקות הוגנות.

2) **הגבלת זמן –** חישוב החלוקה ההוגנת ביותר במסגרת הזמן שלרשותנו.

3) **מהירות –** חישוב החלוקה ההוגנת ביותר בזמן קצר פי 4

תרגיל כיתה

נתון מערך a של 100 מיליון מספרים ממשיים.

צריך לחשב את סכום המספרים בשלישית:

$$\sum_{i=0}^{a.length-1} (a[i])^3$$

נניח שהחישוב הסדרתי לוקח 8 שניות.

כיתבו תוכנית עם שני חוטים המחשבת ב-4 שניות.

פתרונות

- שיטת יצרן-צרכן (Producer-Consumer):
  - כל חוט מחשב את הסכום בחצי מהמערך ומכניס אותו לתור – BlockingQueue.
  - התוכנית הראשית מחכה שהחוטים יסתיימו – join, ואז מוציאה את הסכומים מהתור ומחברת אותם.
- שיטת מיפוי-צמצום (map-reduce):
  - הופכים את המערך לזרם (stream);
  - משתמשים ב-map כדי למפות כל ערך לערך בשלישית;
  - משתמשים ב-reduce כדי לחשב את סכום כל הערכים.