

**046273 - תכנות פונקציונלי ומבוזר**

**תרגיל בית 2: תכנות מקבילי**

תאריך הגשה מפורסם ב-Moodle

## תרגיל ראשון – שרת לחישוב מקבילי של כפל מטריצות

בתרגיל זה תתכננו ותממשו שרת אשר מבצע פעולות חישוב של כפל מטריצות בצורה מקבילית.

ע"מ לאפשר עבודה מקבילית, כפל המטריצות יתבצע בתהליך נפרד מתהליך השרת הראשי. תהליך הכפל ייצור תהליך עבור כל אחד מאיברי המטריצה המוחזרת. (כל תהליך משנה יבצע כפל וקטורים). כאשר כל תהליכי המשנה יסיימו (רמז: ניתן לספור את הודעות הסיום במשתנה מצב) ישלח תהליך הכפל את התוצאה ישירות ללקוח.

ניתן להניח שכל התהליכים, פרט לתהליך השרת ראשי, אינם נופלים לפני סיום עבודתם וחסונים לתקלות. על תהליך השרת הראשי עליכם להגן מפני נפילות.

השרת יחזיק מזהה גרסה מקודד באופן קשיח (hard coded). מזהה הגרסה הינו version\_1.

מטריצות נשמרות בצורת tuple של שורות שכל שורה היא tuple של איברים. לדוגמא, המטריצה  $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$  מיוצגת ע"י  $\{\{1,2,3\},\{4,5,6\},\{7,8,9\}\}$ .

**השרת צריך לתמוך בארבעת ההודעות הבאות (בקשר עם הלקוח):**

1. בקשת ביצוע כפל

{Pid, MsgRef, {multiple, Mat1, Mat2}}

ביצוע הכפל ושליחת התוצאה מתבצעים מתהליך נפרד מהתהליך השרת הראשי. מבנה התשובה:

{MsgRef, Mat}

2. בקשת כיבוי

shutdown

אחרי קבלת הודעה זו, השרת לא יקבל יותר בקשות חדשות, אך הבקשות שבטיפול יסתיימו כרגיל.

3. בקשת גרסה

{Pid, MsgRef, get\_version}

מבנה התשובה:

{MsgRef, VersionIdentifier}

שימו לב ש-version\_1 הינו atom קבוע.

4. בקשה לעדכון קוד

sw\_upgrade

השרת יקרא את הקוד מחדש (אפשר להניח שהשינוי היחיד בין הגרסאות הוא מספר הגרסה).

**סה"כ תמיכה ב-4 הודעות.**

ממשו מודול בשם **matrix\_server** אשר יחשוף את הפונקציות הבאות:

start\_server/0

פונקציה אשר תפעיל את תהליך השרת. תהליך השרת ירשם ( register ) תחת השם **matrix\_server**.

shutdown/0

פונקציה אשר תעצור את תהליך השרת.

mult/2

פונקציה אשר מקבלת שתי מטריצות (בפורמט המתואר) ומחזירה את תוצאת הכפל שלהן.

get\_version/0

תחזיר את מספר הגרסה של השרת.

explanation/0 -> { Answer }

מחזירה tuple המכיל את התשובה לשאלה הבאה – "מדוע יש להפריד את המודול של ה-supervisor מהמודול של השרת עצמו?".

## סה"כ 5 פונקציות למימוש.

דגשים נוספים:

- שימו לב כי על המימוש שלכם לתמוך בביצוע פעולות כפל כפי שמתואר בתרגיל בין אם ע"י הפעלת הפונק' הרלוונטיות מהמודול **matrix\_server** או ע"י פנייה לשרת עצמו באופן ישיר באמצעות הודעות.
- ניתן לייצר מודולים נוספים לנוחיותכם. בכל מקרה, מזהה הגרסה של השרת (ה-atom הנתון **version\_1**) צריך להופיע במפורש בתוך המודול המקורי **matrix\_server**. כחלק מהבדיקה יוחלף מזהה הגרסה במזהה חדש ותיבדק יכול טעינת הקוד מחדש.
- הכפלת המטריצות מתבצעת בצורה מקבילית – כלומר עבור כל בקשה לביצוע כפל יש ליצור תהליך חדש. תהליך זה ייצור תהליכים כמספר האיברים במטריצת התוצאה, כל תהליך יהיה אחראי על חישוב איבר אחר. אין ליצור יותר או פחות תהליכים במהלך הפתרון.
- מצורף מודול נוסף לנוחיותכם. ניתן להשתמש בו אך יש לצרפו להגשה.
- ניתן להניח שכל הנתונים חוקיים, ולמטריצות ממדים המאפשרים כפל.

```
-module(matrix).  
-compile(export_all).
```

```
% generate a matrix with X rows and Y columns with zeros
```

```
getZeroMat(X,Y) ->
```

```
list_to_tuple([list_to_tuple([0 || _Y <- lists:seq(1,Y)]) || _X <- lists:seq(1,X)]).
```

```
% return the ROW row of a Matrix in a tuple format
```

```
getRow(Mat,Row) ->
```

```
element(Row,Mat).
```

```
% return the COL col of a Matrix in a tuple format
```

```
getCol(Mat,Col) ->
```

```
list_to_tuple([element(Col,ColData) || ColData <- tuple_to_list(Mat)]).
```

```
% return a new Matrix which is a copy of OldMat with a NewVal as the value of Row,Col
```

```
setElementMat(Row,Col,OldMat, NewVal) ->
```

```
setelement (Row,OldMat ,setelement (Col,element(Row,OldMat),NewVal)).
```

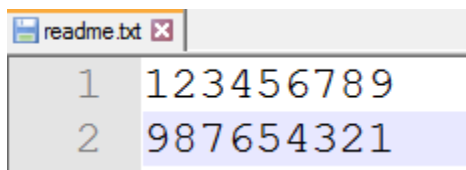
## הוראות הגשה:

- הגשה בזוגות בלבד. ניתן להיעזר בפורום באתר הקורס ע"מ למצוא שותף.
- הגשה אלקטרונית דרך מערכת ה-Moodle. הגשה אחת בלבד עבור שני בני הזוג. שאלות על תוכן התרגיל – דרך הפורום המתאים ב-Moodle.
- עליכם להגיש את כל קבצי המקור (כלומר, .erl) הכוללים תיעוד ומימוש. אין להגיש קבצים מקומפלים (.beam) או תתי-תיקיות.
- לא ניתן להגיש את התרגיל באיחור למעט מקרים פרטניים (מילואים וכו') – בתיאום **מראש** עם המרצה.

בדיקת התרגילים מתבצעת באופן אוטומטי, ולכן עליכם לוודא לפני ההגשה:

- שמות המודולים, הפונקציות והקבועים תואמים לשמות המצוינים בהנחיות התרגיל.
- לא מתבצעות הדפסות לא רצויות למסך.
- הקוד מתקמפל ורץ כהלכה.
- טיפול במקרי קצה – מספרים שליליים ואפס, רשימות ריקות וכו'. מדיניות הטיפול בשגיאות הינה זו שהוגדרה במהלך הקורס - let it crash.

צרו קובץ בשם readme.txt המכיל את ת"ז של שני המגישים בשתי שורות נפרדות. לדוגמא:



```
1 123456789
2 987654321
```

את כל הקבצים יש לארוז בקובץ ZIP (בלבד). שם קובץ ההגשה יהיה

ERLANG\_HW2\_<id1>\_<id2>.zip

כאשר id1 ו-id2 הינם מספרי תעודות הזהות של המגישים. לדוגמא:

ERLANG\_HW2\_123456789\_987654321.zip

לאחר הגשת הקבצים ב-Moodle, אתם נדרשים להוריד את הקבצים שהגשתם ולראות כי הם מתקמפלים ורצים בסביבה נקייה.

הבהרה – אי עמידה בהוראות ההגשה שקולה לאי הגשת התרגיל. אתם נדרשים לוודא כי קיימתם את כל ההוראות המפורטות. לא תהיינה הגשות חוזרות.

## בהצלחה!