



--	--	--	--	--	--	--	--	--

מספר תעודת זהות :

**שם הקורס:** מערכות זמן אמת – תוכנה  
**קוד הקורס:** 10308

**בחינת סמסטר:** א'  
**השנה:** תשע"ה  
**מועד:** ב'  
**מספר ייחוס:** 2

**הוראות לנבחן:**  
- יש לכתוב מס' ת"ז ע"ג טופס הבחינה  
\* יש להחזיר את השאלון עם מחברת הבחינה  
- **אסור כל חומר עזר**, פרט לדפי סיכום של המבחן  
- אסור להעביר מחברות, דפים או ספרים בין נבחנים

**תאריך הבחינה:** 09.03.2015  
**שעת הבחינה:** 17:00  
**משך הבחינה:** 3 שעות  
**מרצה:** מיקי לבנת

- אין לכתוב בעפרון או בצבע אדום  
- בשרטוטים מותר להשתמש בעפרון  
- אין להשתמש בטלפון סלולארי  
- אין להשתמש במחשב אישי או נייד  
- אין להשתמש בדיסק און קי ו/או מכשיר מדיה אחר

### מבנה הבחינה והנחיות לפתרון:

- משקל כל שאלה זהה – 25 נקודות.
- יש לענות על 4 מתוך 5 השאלות .
- במקרה שענית על יותר – יבדקו רק 4 הראשונות.
- יש למחוק שאלות שאינכם רוצים שייבדקו או לציין בתחילת הבחינה איזה שאלות יש לבדוק.
- חלקי שאלות שונות לא יאוחדו לשאלה שלמה.
- שאלה 1 יש לענות על גבי השאלון
- יש להקיף בעיגול את מס' סעיף התשובה הנכונה **בטופס הבחינה**.
- שאר השאלות - יש לענות רק במחברת הבחינה ולא על גבי הטופס.
- יש להתחיל כל שאלה **בראש עמוד חדש**.

**בהצלחה!**



שאלות אמריקאיות – יש להקיף בעיגול את מס' סעיף התשובה הנכונה על השאלון (25 נק)

- A. אלגוריתם EF (Execution-time First) מקנה עדיפות למשימות לפי זמן ביצוע. ככל שזמן ביצוע המשימה גדול יותר, עדיפותה גדולה יותר. באיזה מהאלגוריתמים הבאים אין שינוי בתיזמון מערכת מסוג Preemptive ו-Non-Preemptive? (הכוונה לכל מערכת ולא לדוגמה ספציפית):
- אלגוריתם RM
  - אלגוריתם DM
  - אלגוריתם QF
  - אלגוריתם EF
  - א+ב נכונות
  - א+ג נכונות
  - ג+ד נכונות
  - אף תשובה א-ד אינה נכונה

- B. ה-Dispatcher (משגר) אחראי על-
- תיזמון הפעולות בין יחידות המחשב
  - הכנסת משימות לתור וביצוען
  - הכנסת משימות לתור
  - העברת משימות לביצוע
  - העברת משימות לביצוע עפ"י אלגוריתם התור
  - א+ב נכונות
  - א+ב+ג נכונות
  - ב+ד+ה נכונות

- C. באלגוריתמים RM ו-DM:
- אם תיזמון RM לא תקין, גם תיזמון DM לא תקין
  - אם תיזמון DM לא תקין, גם תיזמון RM לא תקין
  - אם תיזמון DM לא תקין, יתכן שתיזמון RM יהיה תקין
  - אם תיזמון DM תקין, גם תיזמון RM תקין
  - אף תשובה א-ד אינה נכונה



D. היפרמחזור של מערכת משימות מחזוריות : (יותר מתשובה אחת נכונה)

- א. תלוי בזמני ביצוע של המשימות
- ב. תלוי ב- Deadline של המשימות
- ג. תלוי במחזוריות המשימות
- ד. תלוי בזמני השיגור של המשימות
- ה. תלוי במספר המשימות במערכת
- ו. יכול להשתנות אם המערכת היא מסוג Preemptive או Non-Preemptive
- ז. יכול להשתנות אם המערכת היא סטטית או דינמית

E. במערכת Rate Monotonous סכום הנצילויות של המשימות צריך להיות קטן או שווה מערך

מסויים התלוי במספר המשימות. נסמן ערך זה באות W. הנכון לגבי W :

- א. אם הוא קטן מסכום הנצילויות של המשימות, אזי קיים בוודאות שיגור RM תקין
- ב. אם הוא גדול מסכום הנצילויות של המשימות, אזי קיים בוודאות שיגור RM תקין
- ג. אם הוא גדול מסכום הנצילויות של המשימות, אזי יתכן שקיים שיגור RM תקין
- ד. אם הוא קטן מסכום הנצילויות של המשימות, אזי יתכן שקיים שיגור RM תקין
- ה. א+ג נכונות
- ו. א+ד נכונות
- ז. ב+ג נכונות
- ח. ב+ד נכונות



מערכת זמן אמת משתמשת בשעון היוצר בקשת פסיקה כל 20 מילישניות.  
במערכת פועלת תכנית אחת פרימיטיבית המתוארת בקוד האלגוריתמי הבא:

```
Void Interrupt isr(...)
{
    קוד שזמן ביצועו הוא 25 מילישניות [*]
    Return from interrupt
}
```

```
Void Main()
{
    קבל פסיקת isr
    המתן ללחיצת מקש כלשהו
}
```

- 8 נק') א. האם המערכת תשרת את כל בקשות הפסיקה? נמק'!!
- 12 נק') ב. אם זמן ביצוע הקוד (המסומן ב- [\*]), הוא 50 מילישניות, אך צריך להתבצע רק כל  $\frac{1}{2}$  שניה, האם ניתן לשפר את ביצועי המערכת, אם נהפוך אותה למערכת חזית/רקע (Foreground/Background)? אם כן, הסבר כיצד.
- 5 נק') ג. מה נבדק לפני קבלת פסיקה במערכת? מי בודק זאת?



### שאלה מס' 3

במערכת זמן אמת פועלות שלוש משימות  $P_1, P_2$  ו- $P_3$  על CPU's נפרדים, ושני משאבים  $CR_1$  ו- $CR_2$  המשותפים לשלוש המשימות. המשימות פועלות על CPU's נפרדים, ומשתמשות בשישה קטעי קוד לפניה אל המשאבים המשותפים כפי שנראה בטבלה הבאה:

משימה משאב	$P_1$	$P_2$	$P_3$
$CR_1$	$CS_{11}$	$CS_{21}$	$CS_{31}$
$CR_2$	$CS_{12}$	$CS_{22}$	$CS_{32}$

העבודה עם המשאבים המשותפים נעשית בעזרת סמפורים בינאריים  $S_1$  ו- $S_2$  ופקודות WAIT ו-SIGNAL. כמו כן משתמשות המשימות בקטעי הקוד  $NCS_1, NCS_2$  ו- $NCS_3$  שאינן פונות אל המשאבים.  
הערה: ראשי התיבות NCS ו- $CS$  מציינים Non Critical Section ו-Critical Section בהתאמה.

המשימות מתוארות באופן הבא:

P1:	P2:	P3:
WHILE(1)	WHILE(1)	WHILE(1)
{	{	{
1 $NCS_1$	8 $NCS_2$	15 $NCS_3$
2 WAIT $S_1$	9 WAIT $S_2$	16 WAIT $S_1$
3 $CS_{11}$	10 $CS_{22}$	17 WAIT $S_2$
4 WAIT $S_2$	11 WAIT $S_1$	18 $CS_{31}$
5 $CS_{12}$	12 $CS_{21}$	19 SIGNAL $S_2$
6 SIGNAL $S_2$	13 SIGNAL $S_1$	20 $CS_{32}$
7 SIGNAL $S_1$	14 SIGNAL $S_2$	21 SIGNAL $S_1$
}	}	}

- א. (5 נק') הגדר סמפור והפעולות שניתן לבצע עליו. (באופן כללי)
- ב. (7 נק') הראה שיתכן מצב שבו המערכת לא תפעל באופן תקין! השתמש במספרי הפקודות כדי לתאר את סדר הפקודות שיבוצעו והסבר מה יקרה כתוצאה מכך.
- ג. (8 נק') שנה את סדר הפקודות כך שהמערכת תוכל לפעול תמיד באופן תקין – גם אם על חשבון זמן ביצוע ארוך יותר!
- ד. (5 נק') באיזה מקרה תיזמון המשימות כפי שנתון בשאלה יצליח תמיד?



There is a system with 5 tasks  $f_1, f_2, f_3, f_4, f_5$ .

Each task operates on a separate CPU. There is a critical resource for all the tasks.

The tasks are entered in the following order:

$f_1, f_2, f_3, f_4, f_5$   
1 3 2 3 1

All the CPU's operate in the same speed.

The tasks scheduler operates according to LAMPORT Protocol.

(20 pts) a. Complete the following table with the scheduled tasks of the system.

Exspression	J	NUM[i]	Task and its state in the system

(5pts) b. What is the purpose of the "choosing" variable in Lamport Protocol?

שאלה מס' 5

בטבלה הבאה מופיעים הנתונים של 3 משימות מחזוריות :

Task	Period	Execution time	Deadline
T1	4	1	3
T2	5	2	4
T3	10	3	8

6 נק') א. הסבר את עקרונות אלגוריתם EDF. מה ההבדלים בין EDF ל-DM?

5 נק') ב. האם ניתן לומר שהמערכת תקינה לפי אלגוריתם RM מבלי לתזמנה.

10 נק') ג. שרטט תיזמון המערכת לפי אלגוריתם EDF.

4 נק') ד. בצע בדיקת תקינות למערכת.



תקציר קורס מערכות זמן אמת (10308) למבחן

הערות:

- הסיכום אינו מיועד ללמוד ממנו את החומר אלא להיעזר בו לתזכורת על החומר שחזרתם עליו לפי רשימות הכיתה ותרגילים

**כל החומר של השעורים מחייב**

סמפורים:

פעולות על סמפורים:

1. אם $S > 0$ אז הקטן ב- 1 והמשך בתוכנית	Wait	P(S)
2. אם $S = 0$ המתן עד מצב ש $s > 0$ ועבור לסעיף 1		
הגדל את S ב- 1	Post	V(S)

פרוטוקול Lamport

/\* Setting Initial Settings \*/

num[i]=0  $1 \leq i \leq N$

choosing[i]=0  $1 \leq i \leq N$

$P_i$ :

```

.
. NCSi
.
choosing[i]=true
num[i]=1+max(num[1],num[2],...,num[N])
choosing[i]=false
    
```

for j=1 to N

L: if (choosing[j]=true) then goto L

M: if (num[j]!=0)and((num[j],j)<(num[i],i)) then goto M

CS<sub>i</sub>

num[i]=0

```

.
. NCSi
    
```

Goto Pi



תקציר קורס מערכות זמן אמת (10308) למבחן

מושגים בסיסיים בשיגור משימות

Job	יח' עבודה המשוגרת ומבוצעת ע"י המערכת, מתבצע ע"י המערכת ההפעלה ולפעל על משאבי המערכת.
Task	אוסף של job-ים שמבצעים פונקציונאליות מסוימת.
Release Time – RT	הזמן המוקדם ביותר שבו job רצוי להתבצע. הקוד של job נמצא כבר בדרך כלל בזיכרון המערכת במצב Dormant ועובר ל- Ready רק כאשר הגיע הזמן שלו. הזמן הזה הוא לא בהכרח הזמן שהjob מתחיל להתבצע בפועל, אלה תלוי באלגוריתם התור.
Deadline – D	הזמן המאוחר ביותר שבו job חייב לסיים ביצוע.
Period – P - מחזור	פרק זמן שבין ה- RT (Release Time) של 2 job-ים עוקבים של אותה משימה המתבצעים זה אחר זה.
Execution Time - E – זמן ביצוע	פרק הזמן הקצר ביותר שלוקח לjob להתבצע כאשר כל המשאבים עומדים לרשותו והוא מתבצע ברצף.
Utilization – נצילות	$U_i = \frac{E_i}{P_i} ; U = \frac{E}{P} = \frac{\text{Execution Time}}{\text{Period}}$
היפר מחזור	כפולה משותפת מנימלית של מחזורי המשימות. $\#I = H \cdot (1 - U_T) ; N_i = \frac{H}{P_i}$

סוגי Deadline

יחסי	הזמן בין D (Deadline) אבסולוטי לRT (Release Time) של אותו job. קבוע.
אבסולוטי	הזמן בפועל בו job חייב לסיים ביצוע על ציר הזמן. משתנה.
אפקטיבי	גיוב שאין לו גיובים עוקבים (המתבצעים אחריו) שווה לD המוחלט שלו. אם יש גיובים המתבצעים אחריו, ה- D האפקטיבי שלו הוא הערך המינימאלי מתוך ה- D שלו עצמו וכל ערכי ה- D של אלו המתבצעים אחריו.

סוגי Release-Time

נתון	הנתון של המשימה
אפקטיבי	גיוב שאין לו גיובים המתבצעים לפניו שווה ל- RT המוחלט שלו. אם קיימים גיובים המתבצעים לפניו, ה- RT האפקטיבי הוא הערך המקסימאלי של ה- RT שלו עצמו וכל ערכי ה- RT הקודמים לו.



## תקציר קורס מערכות זמן אמת (10308) למבחן

שם	תאור
Clock-Driven עם חלוקה ל-Frames	כללים החלים על Frames : 1. $f \geq \max(E_i)$ 2. מספר שלם $H/f =$ 3. $2f - \gcd(P_i, f) \leq D_i$
RM	מקנה עדיפויות עפ"י מחזוריות הקטנה ביותר
DM	מקנה עדיפויות עפ"י Deadline היחסי הקטן ביותר
QF	מקנה עדיפויות עפ"י סדר כניסת המשימות לתור
EDF	מקנה עדיפויות עפ"י Deadline האבסולוטי הקטן ביותר
LST	מקנה עדיפויות עפ"י ה-Slack הקטן ביותר. $Slack = D - t - x = d - x$

### תנאי לתקינות RM

תנאי לכך שמערכת בעלת n משימות בלתי תלויות תהיה ניתנת לתזמון תקין ב-RM בלבד הוא :

$$\sum U \leq n \left( 2^{\frac{1}{n}} - 1 \right)$$

מסקנות :

- אם אי השוויון מתקיים ניתן לומר בוודאות שתזמון RM תקין.
- אם אי השוויון אינו נכון לא ניתן להסיק שום מסקנה. יתכן שהתזמון תקין ויתכן שלא חייבים לבדוק.

**הערה :** התנאי תקין אך ורק עבור  $P \leq D$

כיוון שהחלק השני של התנאי תלוי רק במספר המשימות ניתן ליצור טבלה קבועה.

n	$U_{\max} = n \left( 2^{\frac{1}{n}} - 1 \right)$
1	1
2	0.828
3	0.78
4	0.757
5	0.753
⋮	⋮

### אוטומטים סופיים – מכונת מצבים

אוטומט סופי M הוא פונקציה בעלת 5 פרמטרים :  $M = \{I, Q, q_0, F, f\}$

I	קבוצת קלט האוטומט
Q	קבוצת המצבים האפשריים של האוטומט
$q_0$	מצב התחלתי של האוטומט
F	קבוצת המצבים הסופיים שהאוטומט צריך להגיע אליהם
f	פונקצית המעבר ממצב למצב = טבלת מצבים



פתח



פתרון מואץ ק'  
09.03.2015

**אפקה** המכללה האקדמית להנדסה בתל-אביב  
AFEKA TEL-AVIV ACADEMIC COLLEGE OF ENGINEERING

DEPARTMENT OF  
SOFTWARE ENGINEERING

המחלקה להנדסת  
תוכנה

מספר תעודת זהות : 

--	--	--	--	--	--	--	--	--	--

**שם הקורס:** מערכות זמן אמת – תוכנה - **פתרון**  
**קוד הקורס:** 10308

- |   |  |
|---|--|
| <p><b>הוראות לנבחן:</b></p> <ul style="list-style-type: none"><li>- יש לכתוב מס' ת"ז ע"ג טופס הבחינה</li><li>- יש להחזיר את השאלון עם מחברת הבחינה</li><li>- <b>אסור כל חומר עזר</b>, פרט לדפי סיכום של המבחן</li><li>- אסור להעביר מחברות, דפים או ספרים בין נבחנים</li></ul> <p><b>בחינת סמסטר:</b> א'<br/><b>השנה:</b> תשע"ה<br/><b>מועד:</b><br/><b>מספר ייחוס:</b> 2</p> <p><b>תאריך הבחינה:</b><br/><b>שעת הבחינה:</b> 17:00<br/><b>משך הבחינה:</b> 3 שעות<br/><b>מרצה:</b> מיקי לבנת</p> | <ul style="list-style-type: none"><li>- אין לכתוב בעפרון או <b>בצבע אדום</b></li><li>- בשרטוטים מותר להשתמש בעפרון</li><li>- אין להשתמש בטלפון סלולארי</li><li>- אין להשתמש במחשב אישי או נייד</li><li>- אין להשתמש בדיסק און קי ו/או מכשיר מדיה אחר</li></ul> <p><b>מבנה הבחינה והנחיות לפתרון:</b></p> <ul style="list-style-type: none"><li>• משקל כל שאלה זהה – 25 נקודות.</li><li>• יש לענות על 4 מתוך 5 השאלות .</li><li>• במקרה שענית על יותר – יבדקו רק 4 הראשונות.</li><li>• יש למחוק שאלות שאינכם רוצים שייבדקו או לציין בתחילת הבחינה איזה שאלות יש לבדוק.</li><li>• חלקי שאלות שונות לא יאוחדו לשאלה שלמה.</li><li>• שאלה 1 יש לענות על גבי השאלון</li><li>• יש להקיף בעיגול את מס' סעיף התשובה הנכונה <b>בטופס הבחינה</b>.</li><li>• שאר השאלות - יש לענות רק במחברת הבחינה ולא על גבי הטופס.</li><li>• יש להתחיל כל שאלה <b>בראש עמוד חדש</b>.</li></ul> |
|---|--|

**בהצלחה!**

כל הזכויות שמורות © למיכאל (מיקי) לבנת. אין להעתיק, לצלם, להקליט, לשדר, לאחסן במאגר מידע, בכל דרך שהיא, בין מכאנית ובין אלקטרונית או בכל דרך אחרת, כל חלק שהוא מטופס הבחינה



(25 נק) שאלות אמריקאיות – יש להקיף בעיגול את מס' סעיף התשובה הנכונה על השאלון

- A. אלגוריתם EF (Execution-time First) מקנה עדיפות למשימות לפי זמן ביצוע. ככל שזמן ביצוע המשימה גדול יותר, עדיפותה גדולה יותר. באיזה מהאלגוריתמים הבאים אין שינוי בתיזמון מערכת מסוג Preemptive ו-Non-Preemptive? (הכוונה לכל מערכת ולא לדוגמה ספציפית):
- אלגוריתם RM
  - אלגוריתם DM
  - אלגוריתם QF
  - אלגוריתם EF
  - א+ב נכונות
  - א+ג נכונות
  - ג+ד נכונות
  - אף תשובה א-ד אינה נכונה

- B. ה-Dispatcher (משגר) אחראי על-  
א. תיזמון הפעולות בין יחידות המחשב

- הכנסת משימות לתור וביצוען
- הכנסת משימות לתור
- העברת משימות לביצוע
- העברת משימות לביצוע עפ"י אלגוריתם התור
- א+ב נכונות
- א+ב+ג נכונות
- ב+ד+ה נכונות

- C. באלגוריתמים RM ו-DM:

- אם תיזמון RM לא תקין, גם תיזמון DM לא תקין
- אם תיזמון DM לא תקין, גם תיזמון RM לא תקין
- אם תיזמון DM לא תקין, יתכן שתיזמון RM יהיה תקין
- אם תיזמון DM תקין, גם תיזמון RM תקין
- אף תשובה א-ד אינה נכונה



D. היפרמחזור של מערכת משימות מחזוריות: (יותר מתשובה אחת נכונה)

- א. תלוי בזמני ביצוע של המשימות
- ב. תלוי ב- Deadline של המשימות
- ג. תלוי במחזוריות המשימות
- ד. תלוי בזמני השיגור של המשימות
- ה. תלוי במספר המשימות במערכת
- ו. יכול להשתנות אם המערכת היא מסוג Preemptive או Non-Preemptive
- ז. יכול להשתנות אם המערכת היא סטטית או דינמית

E. במערכת Rate Monotonous סכום הנצילות של המשימות צריך להיות קטן או שווה מערך

מסויים התלוי במספר המשימות. נסמן ערך זה באות W. הנכון לגבי W :

- א. אם הוא קטן מסכום הנצילות של המשימות, אזי קיים בוודאות שיגור RM תקין
- ב. אם הוא גדול מסכום הנצילות של המשימות, אזי קיים בוודאות שיגור RM תקין
- ג. אם הוא גדול מסכום הנצילות של המשימות, אזי יתכן שקיים שיגור RM תקין
- ד. אם הוא קטן מסכום הנצילות של המשימות, אזי יתכן שקיים שיגור RM תקין
- ה. א+ג נכונות
- ו. א+ד נכונות
- ז. ב+ג נכונות
- ח. ב+ד נכונות

## שאלה מס. 2

א. המערכת לא תשרת את כל בקשות הפסיקה, כי זמן הטיפול בפסיקה (25 מ"ש) ארוך מהזמן בין 2 פסיקות עוקבות (20 מ"ש).

ב. אפשר. בהנחה שזמן השירות לפסיקה זנית, אזי כל  $\frac{1}{2}$  שניה תופיע פסיקה אחת. ניתן לממש זאת ע"י דגל או סמפור.

ג. לפני קבלת פסיקה במערכת – מערכת הבקשה בודקת:

- קיים איפשור כל הפסיקות
- קיים איפשור לפסיקה הנוכחית
- לא רצה כרגע במערכת פסיקה בעדיפות גבוהה יותר



### שאלה מס. 3

- א. סמפור S הוא משתנה שיכול לקבל ערכים שלמים חיוביים או 0, ומוגדרות עליו 2 פעולות פרימיטיביות P ו-V המוגדרות כך:
- \*  $P(S) - [i]$  אם  $S > 0$  הקטן אותו באחד והמשך.
  - \*  $[ii]$  אם  $S = 0$  המתן עד שיתקיים  $S > 0$  ואז בצע את  $[i]$ .
  - \*  $V(S) -$  הגדל את S ב-1.
- סמפור יכול להיות בינוי (2 ערכים) או מונה (הרבה ערכים) וחייבים להגדירו ולתת לו תנאי התחלה.
- ב. תנאי התחלה:  $S_1 = S_2 = 1$
- נניח שמשימה  $P_3$  מבצעת ברגע מסוים את  $NCS_3$  ובאותו זמן מבצעות המשימות  $P_1$  את  $CS_{11}$  ו- $P_2$  את  $CS_{22}$ . עם סיום  $CS_{11}$  ו- $CS_{22}$  נחסמות  $P_2$  ו- $P_1$  ע"י הסמפורים  $S_2$  ו- $S_1$  בהתאמה וכך גם  $P_3$  תחסם ע"י  $S_1$ . זהו מצב של DEADLOCK.
- ג. ניתן לפתור את הבעיה ע"י שינוי סדר הפקודות באופן הבא:

P1:		P2:		P3:
WHILE(1)		WHILE(1)		WHILE(1)
{		{		{
1 NCS <sub>1</sub>	8	NCS <sub>2</sub>	15	NCS <sub>3</sub>
2 WAIT S <sub>1</sub>	9	WAIT S <sub>2</sub>	16	WAIT S <sub>1</sub>
3 CS <sub>11</sub>	10	CS <sub>22</sub>	17	CS <sub>31</sub>
4 SIGNAL S <sub>1</sub>	11	SIGNAL S <sub>2</sub>	18	SIGNAL S <sub>1</sub>
5 WAIT S <sub>2</sub>	12	WAIT S <sub>1</sub>	19	WAIT S <sub>2</sub>
6 CS <sub>12</sub>	13	CS <sub>21</sub>	20	CS <sub>32</sub>
7 SIGNAL S <sub>2</sub>	14	SIGNAL S <sub>1</sub>	21	SIGNAL S <sub>2</sub>
}		}		}

S1 מגן על CR1 ו-S2 מגן על CR2.

- ד. אם המשימות ירוצו על CPU יחיד ב"טור", הן לעולם לא תיקענה והתיזמון יהיה תקין.  
אך זאת לא מערכת זמן אמת.



א.

 $f_1, f_2, f_3, f_4, f_5$   
 1 3 2 3 1

משימה ומצבה במערכת	NUM[i]	J	הביטוי
<u>f 1 מתחילה ונכנסת ל- CS1</u>	NUM[1]=1		
f 5 מתחילה ונתקעת ב- M	NUM[5]=1	1	(1,1) < (1,5)
f 3 מתחילה ונתקעת ב- M	NUM[3]=2	1	(1,1) < (3,2)
f 2 מתחילה ונתקעת ב- M	NUM[2]=3	1	(1,1) < (3,2)
f 4 מתחילה ונתקעת ב- M	NUM[4]=3	1	(1,1) < (3,4)
f 1 מסיימת את CS3	NUM[1]=0	----	----
<u>f 5 משתחררת מ-M ונכנסת ל- CS5</u>	NUM[5]=1		NUM[1]=0
f 3 נשארת תקועה ב- M	NUM[3]=2	5	(1,5) < (2,3)
f 2 נשארת תקועה ב- M	NUM[2]=2	3	(2,3) < (3,2)
f 4 נשארת תקועה ב- M	NUM[4]=2	2	(2,2) < (2,4)
f 5 מסיימת את CS5	NUM[5]=0	----	----
<u>f 3 משתחררת מ-M ונכנסת ל- CS3</u>	NUM[3]=2		NUM[5]=0
f 2 נשארת תקועה ב- M	NUM[2]=2	3	(2,3) < (3,2)
f 4 נשארת תקועה ב- M	NUM[4]=2	2	(2,2) < (2,4)
f 3 מסיימת את CS1	NUM[3]=0	----	----
<u>f 2 משתחררת מ-M ונכנסת ל- CS2</u>	NUM[2]=2		NUM[3]=0
f 4 נשארת תקועה ב- M	NUM[4]=2	2	(2,2) < (2,4)
f 2 מסיימת את CS2	NUM[2]=0	----	----
<u>f 4 משתחררת מ-M ונכנסת ל- CS4</u>	NUM[4]=2		NUM[2]=0
f 4 מסיימת את CS4	NUM[4]=0	----	----



ב. ה-choosing נועד למנוע מעבר משורה L לשורה M אם אחת המשימות מבצעת עידכון ה- NUM(i) שלה בשורה K. לפיכך בזמן זה  $choosing(j) = false$  ותוקע את הפרוטוקול בשורה L עד שה-choosing של המשימה יעבור ל-true (המשימה סיימה לעדכן את NUM(i) שלה).

### שאלה מס. 5

בטבלה הבאה מופיעים הנתונים של 3 משימות מחזוריות :

Task	Period	Execution time	Deadline
T1	4	1	3
T2	5	2	4
T3	10	3	8

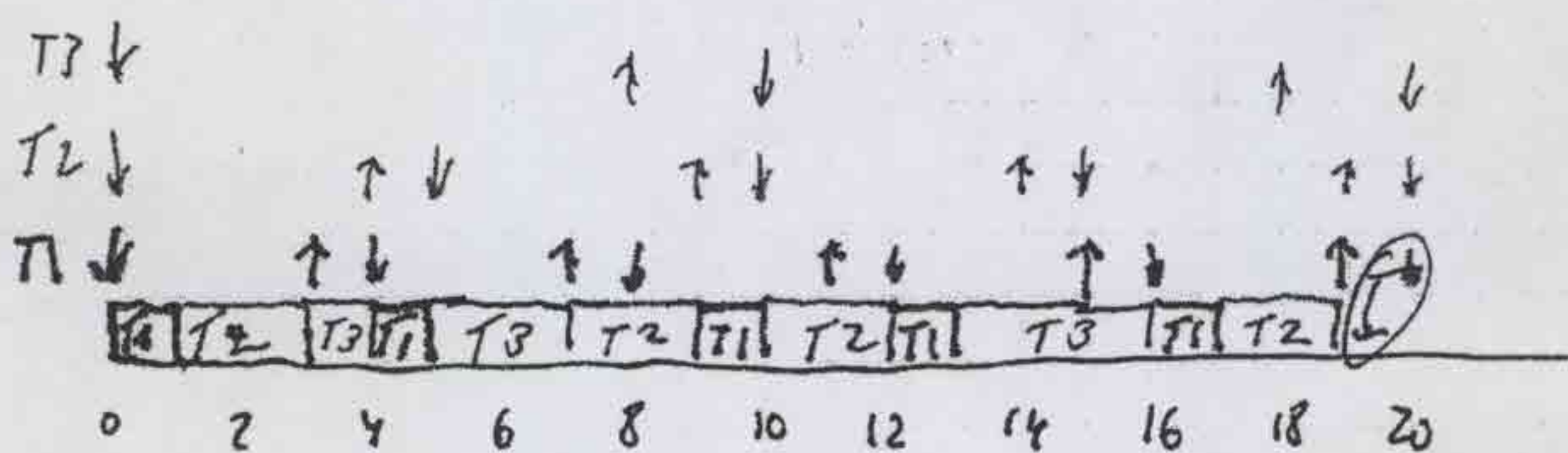
א. EDF הוא אלגוריתם עם עדיפויות דינמיות, שנותן עדיפות למשימה שה-Deadline האבסולוטי שלה הוא הקרוב ביותר בזמן הבדיקה. בודקים כאשר משימה נכנסת לתור או מסיימת ביצוע.

DM הוא אלגוריתם עם עדיפויות סטטיות, שנותן עדיפות למשימה עם ה-Deadline היחסי הקטן ביותר.

הבדלים: DM סטטי, EDF דינמי. משימה ב-DM יכולה להתחיל רק בתחילת Clock. משימה ב-EDF יכולה להתחיל להתבצע גם באמצע Clock.

ב. היות ולא מתקיים  $P \leq DL$  לא ניתן להשתמש בנוסחא של RM ולכן לא ניתן לדעת, ללא תיזמון.

ג.



$$\#I = H(1 - U_T) = 20(1 - 0.95) = 1$$

$$N_3 = 20/10 = 2 ; N_2 = 20/5 = 4 ; N_1 = 20/4 = 5 ; N_i = H/P_i$$