



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

מספר תעודת זהות :

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

בחינת סמסטר: א'
השנה: תשע"ד
מועד: X'
מספר ייחוס: 1

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: 31.01.2014
שעת הבחינה: 8:30
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**.

--

מס' נבחן

בהצלחה!

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

- A. באלגוריתם RM קיים קריטריון לבדיקת תקינות ללא תיזמון:
- אם הקריטריון מצליח, המערכת בוודאות תקינה ב-RM למערכת Non-Preemptive בלבד
 - אם הקריטריון מצליח, המערכת בוודאות תקינה ב-RM למערכת Preemptive בלבד
 - אם הקריטריון מצליח, ו- $D < P$ באחת המשימות, המערכת בוודאות תקינה ב-RM
 - אם הקריטריון מצליח, ו- $D \geq P$ באחת המשימות, המערכת בוודאות תקינה ב-RM
 - אם הקריטריון מצליח, ו- $D \geq P$ בכל המשימות, המערכת בוודאות תקינה ב-RM
 - אם הקריטריון מצליח, המערכת בוודאות תקינה עבור RM ו-DM
 - אם הקריטריון אינו מצליח, המערכת אינה תקינה עבור RM
 - אף תשובה א-ז אינה נכונה

- B. אלגוריתם EF (Execution Time First) מקנה עדיפויות למשימות לפי זמן הביצוע שלהן. משימה שזמן הביצוע שלה גדול יותר, מקבלת עדיפות גבוהה יותר. אלגוריתם זה: (יותר מתשובה אחת נכונה)
- הוא עם עדיפויות דינמיות
 - הוא עם עדיפויות סטטיות
 - תיזמון Preemptive ו-Non-Preemptive יכול להיות שונה
 - תיזמון Preemptive ו-Non-Preemptive תמיד זהה
 - לא יכול להיות חריגות בתיזמון, כי תמיד זמן ביצוע קטן מ-Deadline במערכות

- C. לגבי Release Time של משימה: (יותר מתשובה אחת נכונה)
- הוא פאזה (Phase) של המשימה
 - Release Time אפקטיבי מתאים למשימות בלתי תלויות
 - Release Time אפקטיבי מתאים למערכת מרובת CPU's
 - לפני מועד ה-Release Time, המשימה נמצאת בתור
 - לפני מועד ה-Release Time, המשימה נמצאת ב-Dormant
 - Release Time יחסי משתנה בעת התיזמון של המערכת

D. הנכון Preemptive ו- Non-Preemptive : (יותר מתשובה אחת נכונה)

- א. המונח Non-Preemptive ו- Preemptive משפיע על משימות שנמצאות בתור
- ב. המונח Non-Preemptive ו- Preemptive משפיע על משימות שנמצאות ב- Running
- ג. במערכת מסוג Preemptive, משימה בעדיפות גבוהה שנכנסת לתור, מפסיקה ביצוע של משימה בעדיפות נמוכה יותר
- ד. במערכת מסוג Preemptive, משימה בעדיפות גבוהה שנכנסת ל- Dormant, מפסיקה ביצוע של משימה בעדיפות נמוכה יותר
- ה. במערכת מסוג Non-Preemptive, קיימת אפשרות שמשימה תופסק ע"י משימה בעדיפות גבוהה

E. זמן חילופי קשר הוא : (יותר מתשובה אחת נכונה)

- א. זהו הזמן הדרוש למשימה שהופסקה לעבור למצב Ready
- ב. זהו הזמן הדרוש למשימה שהופסקה לעבור למצב ISR
- ג. זהו הזמן הדרוש למשימה שהופסקה לעבור למצב Ready ועד שהמשימה החדשה מתחילה לעבוד
- ד. זהו הזמן הדרוש למשימה שהופסקה לעבור למצב ISR ועד שהמשימה החדשה מתחילה לעבוד
- ה. קיים רק במשימות מסוג Reentrant
- ו. לא קיים במערכת של משימות Non-Preemptive
- ז. אף תשובה א-ו אינה נכונה

שאלה מס' 2

בטבלה הבאה מופיעים הנתונים של 3 משימות מחזוריות:

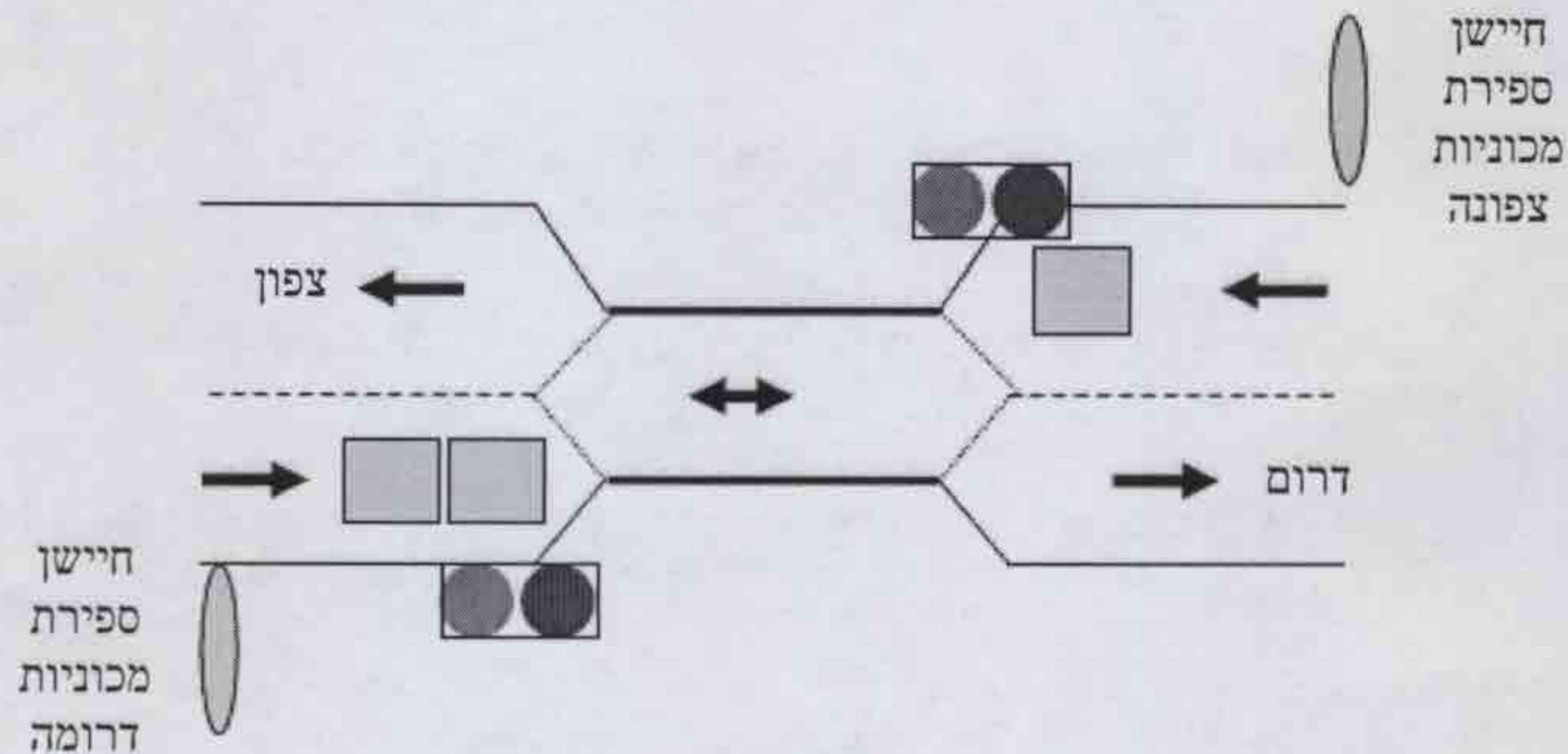
משימה	זמן מחזור	זמן ביצוע
T1	6	2
T2	12	4
T3	4	1

לכל המשימות יש מופע 0 ו- $\text{Period} = \text{Deadline}$

- א. (5 נק') האם ניתן לתזמן את המערכת לפי אלגוריתם RM? האם לפי DM? (ללא חישוב דיאגרמת הזמנים)
- ב. (10 נק') תזמן את המערכת עבור Preemptive DM. האם התזמון הצליח? אם לא ציין היכן נכשל.
- ג. (10 נק') תזמן את המערכת עבור Non-Preemptive DM. האם התזמון הצליח? אם לא ציין היכן נכשל.

שאלה מס' 3

קיים גשר צר המאפשר תנועת מכוניות בכיוון אחד בלבד (צפונה/דרומה), עפ"י האזור הבא. בכניסה לגשר יש רמזור מכל כיוון שדולק באור אדום או ירוק. לפני הגשר בכל כיוון קיימים חיישנים הסופרים את מס' המכוניות שנעצרות לפני הרמזור. אם נעמדת מכונית המס' גדל באחד. אם עוברת מכונית המס' קטן באחד. רק רמזור אחד יכול להיות ירוק בו זמנית והשני חייב להיות אדום. כאשר אין אף מכונית ממתונה ב- 2 הכיוונים, יש לדאוג שהרמזורים יתחלפו לירוק בהתאמה כל 45 שניות.



כתוב אלגוריתם תוך שימוש בסמפורים כרצונך, לוויסות התנועה בגשר. מכונית תמתין מקסימום 5 מכוניות שעוברות לכיוון הנגדי, עד שתקבל הרשאה (אור ירוק). הערות:

1. החיישנים סופרים את מס' המכוניות הממתונות אוטומאטית והערכים שלהם נתונים במשתנים:
 - N_N – מס' מכוניות ממתונות צפונה
 - N_S – מס' מכוניות ממתונות דרומה
2. בנוסף לסמפורים, אם רוצים, ניתן להשתמש גם בלולאות (כמו for, if וכו').
3. במחזור של 45 שניות, לא מתחשבים בתורי המכוניות, עד לסיומו.

שאלה מס' 4

בטבלה הבאה מופיעים הנתונים של 3 משימות מחזוריות :

משימה	מחזור	זמן ביצוע
T_1	8	2
T_2	6	3
T_3	2	0.5

לכל המשימות יש מופע 0 ו- $d=P$.

- א. (3 נק') הסבר את עקרון הפעולה של אלגוריתם (LST) Least Slack Time. מה יתרון LST ע"פ EDF (Earlier Deadline First)?
- ב. (15 נק') שרטט Gantt Chart (דיאגרמת זמנים) של המערכת בשיטת LST, עבור היפרמחזור שלם, למערכת מסוג Preemptive.
- ג. (8 נק') מה קורה אם ל-2 משימות יש אותו ערך של Slack בזמן מסויים? האם המערכת הגיונית? מה היית משנה?

שאלה מס' 5

בטבלה הבאה מופיעים הנתונים של 3 משימות מחזוריות :

משימה	מחזור	זמן ביצוע	DEADLINE
T_1	5	2	4
T_2	10	3	6
T_3	15	4	11

לכל המשימות יש מופע 0 (זמן השיגור של התהליך הראשון של התהליך הראשון של כל משימה הוא 0).

- א. (6 נק') מה היתרון של EDF על-פני DM? מה החיסרון של EDF לעומת LST?
- ב. (6 נק') הסבר כיצד פועל אלגוריתם EDF.
- ב. (13 נק') שרטט Gantt Chart (דיאגרמת זמנים) של המערכת בשיטת EDF (Earlier Deadline First) עבור היפרמחזור שלם.

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 */

$\text{num}[i]=0 \ 1 \leq i \leq N$

$\text{choosing}[i]=0 \ 1 \leq i \leq N$

P_i :

. NCS_i

. $\text{choosing}[i]=\text{true}$

$\text{num}[i]=1+\max(\text{num}[1], \text{num}[2], \dots, \text{num}[N])$

$\text{choosing}[i]=\text{false}$

for $j=1$ to N

L: if ($\text{choosing}[j]=\text{true}$) then goto L

M: if ($\text{num}[j] \neq 0$) and ($(\text{num}[j], j) < (\text{num}[i], i)$) then goto M

CS_i

$\text{num}[i]=0$

. NCS_i

Goto P_i

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, אלא 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 מינימלית של מחזורי המשימות. $\#I = H(1 - U_T) ; N_i = \frac{H}{P_i}$

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

אלגוריתמים

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 :

1. אם אי השוויון מתקיים ניתן לומר בוודאות שתזמון RM תקין.
2. 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	פונקציית המעבר ממצב למצב = טבלת מצבים

פתח

פתרון
מאגרי א'
31.01.2014

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

מספר תעודת זהות :

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

- | | |
|---|---------------------------|
| הוראות לנבחן: | בחינת סמסטר: א' |
| - יש לכתוב מס' ת"ז ע"ג טופס הבחינה | השנה: תשע"ד |
| - יש להחזיר את השאלון עם מחברת הבחינה | מועד: X |
| - אסור כל חומר עזר , פרט לדפי סיכום של המבחן | מספר ייחוס: 1 |
| - אסור להעביר מחברות, דפים או ספרים בין נבחנים | |
| - אין לכתוב בעפרון או בצבע אדום | תאריך הבחינה: |
| - בשרטוטים מותר להשתמש בעפרון | שעת הבחינה: 8:30 |
| - אין להשתמש בטלפון סלולארי | משך הבחינה: 3 שעות |
| - אין להשתמש במחשב אישי או נייד | מרצה: מיקי לבנת |
| - 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!

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

- A. באלגוריתם RM קיים קריטריון לבדיקת תקינות ללא תיזמון:
- אם הקריטריון מצליח, המערכת בוודאות תקינה ב-RM למערכת Non-Preemptive בלבד
 - אם הקריטריון מצליח, המערכת בוודאות תקינה ב-RM למערכת Preemptive בלבד
 - אם הקריטריון מצליח, ו- $D < P$ באחת המשימות, המערכת בוודאות תקינה ב-RM
 - אם הקריטריון מצליח, ו- $D \geq P$ באחת המשימות, המערכת בוודאות תקינה ב-RM
 - אם הקריטריון מצליח, ו- $D \geq P$ בכל המשימות, המערכת בוודאות תקינה ב-RM
 - אם הקריטריון מצליח, המערכת בוודאות תקינה עבור RM ו-DM
 - אם הקריטריון אינו מצליח, המערכת אינה תקינה עבור RM
 - אף תשובה א-ז אינה נכונה

B. אלגוריתם EF (Execution Time First) מקנה עדיפויות למשימות לפי זמן הביצוע שלהן.

משימה שזמן הביצוע שלה גדול יותר, מקבלת עדיפות גבוהה יותר.

אלגוריתם זה: (יותר מתשובה אחת נכונה)

- הוא עם עדיפויות דינמיות
- הוא עם עדיפויות סטטיות
- תיזמון Preemptive ו-Non-Preemptive יכול להיות שונה
- תיזמון Preemptive ו-Non-Preemptive תמיד זהה
- לא יכול להיות חריגות בתיזמון, כי תמיד זמן ביצוע קטן מ-Deadline במערכות

C. לגבי Release Time של משימה: (יותר מתשובה אחת נכונה)

- הוא פאזה (Phase) של המשימה
- Release Time אפקטיבי מתאים למשימות בלתי תלויות
- Release Time אפקטיבי מתאים למערכת מרובת CPU's
- לפני מועד ה-Release Time, המשימה נמצאת בתור
- לפני מועד ה-Release Time, המשימה נמצאת ב-Dormant
- Release Time יחסי משתנה בעת התיזמון של המערכת

D. הנכון Preemptive ו- Non-Preemptive : (יותר מתשובה אחת נכונה)

- א. המונח Preemptive ו- Non-Preemptive משפיע על משימות שנמצאות בתור
- ב. המונח Preemptive ו- Non-Preemptive משפיע על משימות שנמצאות ב- Running
- ג. במערכת מסוג Preemptive, משימה בעדיפות גבוהה שנכנסת לתור, מפסיקה ביצוע של משימה בעדיפות נמוכה יותר
- ד. במערכת מסוג Preemptive, משימה בעדיפות גבוהה שנכנסת ל- Dormant, מפסיקה ביצוע של משימה בעדיפות נמוכה יותר
- ה. במערכת מסוג Non-Preemptive, קיימת אפשרות שמשימה תופסק ע"י משימה בעדיפות גבוהה

E. זמן חילופי קשר הוא : (יותר מתשובה אחת נכונה)

- א. זהו הזמן הדרוש למשימה שהופסקה לעבור למצב Ready
- ב. זהו הזמן הדרוש למשימה שהופסקה לעבור למצב ISR
- ג. זהו הזמן הדרוש למשימה שהופסקה לעבור למצב Ready ועד שהמשימה החדשה מתחילה לעבוד
- ד. זהו הזמן הדרוש למשימה שהופסקה לעבור למצב ISR ועד שהמשימה החדשה מתחילה לעבוד
- ה. (לא נכון, כי משימה לא עוברת ל- ISR ע"י משימה)
- ו. קיים רק במשימות מסוג Reentrant
- ז. לא קיים במערכת של משימות Non-Preemptive
- ח. אף תשובה א-ו אינה נכונה

שאלה מס' 2

א. עבור RM : לא מתקיים $UT \leq U_{max}$ ולכן אי אפשר לדעת.

עבור DM : מאחר ו- $Period = Deadline$, תיומון $DM = RM$ וגם לא ניתן לדעת.

ב. $H=12$

Preemptive

T3	T1	T1	T2	T3	T2	T1	T1	T3	T2	T2	IDLE	IDLE
0	1	2	3	4	5	6	7	8	9	10	11	12

התזמון מצליח.

ג. Non-Preemptive

T3	T1	T1	T2	T2	T2	T2	T3	T3	T1	T1	IDLE	IDLE
0	1	2	3	4	5	6	7	8	9	10	11	12

התזמון מצליח.

שאלה מס' 3

נגדיר:

- סמפור בינרי $S_R=1$ כסמפור של רמזור ירוק (באחד מהרמזורים. הרמזור השני אוט' הופך לאדום)
- N_N – מס' מכוניות שממתינות צפונה (ערכו מתקבל מהחיישן)
- N_S – מס' מכוניות שממתינות דרומה

<p>NORTH:</p> <pre> P(S_R) if N_N ≥ 5 then goto LOOP if N_S ≥ 5 then V(S_R) //to "GREEN" ramzor south for the waiting cars goto NORTH else ramzor NORTH = "GREEN" endif if N_N = 0 then delay (45 sec) V(S_R) goto NORTH else LOOP: if N_N > 0 goto LOOP //to pass the waiting cars V(S_R) goto NORTH endif </pre>	<p>SOUTH:</p> <pre> P(S_R) if N_S ≥ 5 then goto LOOP if N_N ≥ 5 then V(S_R) //to "GREEN" ramzor north for the waiting car goto SOUTH else ramzor SOUTH = "GREEN" endif if N_S = 0 then delay (45 sec) V(S_R) goto SOUTH else LOOP: if N_S > 0 goto LOOP //to pass the waiting cars V(S_R) goto SOUTH endif </pre>
---	--

NORTH:

```

P(SR)
if NN ≥ 5 then
    goto LOOP
if NS ≥ 5 then
    V(SR) //to "GREEN" ramzor south for the waiting cars
    goto NORTH
else
    ramzor NORTH = "GREEN"
endif
if NN = 0 then
    delay (45 sec)
    V(SR)
    goto NORTH
else
    LOOP: if NN > 0 goto LOOP //to pass the waiting cars
    V(SR)
    goto NORTH
endif
        
```

NORTH:

```

P(SR)
if NN ≥ 5 then
    goto LOOP
if NS ≥ 5 then
    V(SR) //to "GREEN" ramzor south for the waiting cars
    goto NORTH
else
    ramzor NORTH = "GREEN"
endif
if NN = 0 then
    delay (45 sec)
    V(SR)
    goto NORTH
else
    LOOP: if NN > 0 goto LOOP //to pass the waiting cars
    V(SR)
    goto NORTH
endif
        
```

כל הזכויות שמורות © למיכאל (מיקי) לבנת. אין להעתיק, לצלם, לאחסן במאגר מידע, כל חלק שהוא מטופס הבחינה.

שאלה מס' 4

בטבלה הבאה מופיעים הנתונים של 3 משימות מחזוריות:

משימה	מחזור	זמן ביצוע
T_1	4	1
T_2	3	1.5
T_3	1	0.25

לכל המשימות יש מופע 0.

- 5 נק' א. הסבר את עקרון הפעולה של אלגוריתם (LST) Least Slack Time. מה יתרון LST ע"פ EDF Earlier Deadline First?
- 15 נק' ב. שרטט Gantt Chart (דיאגרמת זמנים) של המערכת בשיטת LST, עבור היפרמחזור שלם, למערכת מסוג Preemptive.
- 5 נק' ג. מה קורה אם ל-2 משימות יש אותו ערך של Slack בזמן מסוים? האם המערכת הגיונית? מה היית משנה?

פתרון

LST - Least Slack Time first

א.

אלגוריתם LST (בניגוד ל-EDF ואחרים), לוקח בחישוב עדיפויות המשימות בתור, גם את הזמן שמשימה כבר התבצעה או לא, (נכון יותר את הזמן שנותר לביצוע עבור המשימה). האלגוריתם מחשב בכל רגע החלטה של שיגור את ה-Slack של המשימות, ומקנה עדיפות גבוהה למשימה עם ה-Slack המינימלי, בעת הבדיקה.

$$S = D - t - X = d - X$$

כאשר:

S – ערך ה-Slack

D – Deadline אבסולוטי של גיוב המשימה

D – Deadline יחסי (לזמן הבדיקה) של גיוב המשימה ($D - t =$)

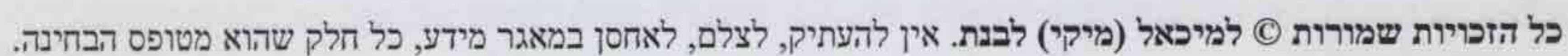
X – הזמן שנותר לביצוע עבור המשימה בעת הבדיקה

t – זמן הבדיקה. בודקים בכל פעם שמשימה משוגרת או מסתיימת.

האלגוריתם עם עדיפות דינמית ומתאים למערכות מסוג Preemptive.

יתרונה יחסית ל-EDF – שמתחשבים בזמנים שמשמיות כבר התבצעו, בזמני הבדיקות.

- ג. במקרה של Slack שווה ל-2 משימות או יותר, העדיפות נקבעת לפי FIFO כניסת משימות למערכת. הבעיה כאן ש- T_3 עם קצב השיגור הגבוה ביותר (כל 2), יש לה עדיפות אחרונה לפי FIFO. לכן LST נכשל ב- $t=16$, כי ב- $t=14$, עדיפויות (Slacks) של T_3 ו- T_2 שוות, והעדיפות ניתנת ל- T_2 עפ"י FIFO. הדבר ההגיוני לשנות, ש- T_3 תכנס ראשונה למערכת. בצורה זו LST יצליח.

H=24

שאלה מס' 5

א. EDF הוא דינמי ונותן עדיפות לפי DL אבסולוטי הקרוב ביותר, לעומת DM סטטי ונותן עדיפות לפי ה-DL היחסי הקטן ביותר.
LST לוקח בחשבון את הזמן שכבר משימה התבצעה ונותן עדיפות למשימה עם ה-Slack המינימלי.

ב. באלגוריתם EDF מעניקים עדיפות למשימה שה-Deadline שלה הקרוב ביותר בזמן הבדיקה t .
בודקים בכל פעם שמשימה משוגרת, או משימה מסיימת ביצוע.

ג. $H=30$

משימה	מחזור	זמן ביצוע	DEADLINE	$U=E/P$
T_1	5	2	4	0.4
T_2	10	3	6	0.3
T_3	15	4	11	0.27
				0.97

