

DEPT OF SYSTEMS ENGINEERING GRADUATE STUDIES

אווודקוה להענטת מעוגות remain line

בחינות

שם הקורס: מערכות הפעלה <u>קוד הקורס:</u> 10303

<u>הוראות לנבחן:</u>

חומר עזר שימושי לבחינה חומר סגור

אין לכתוב בעפרון

אין להשתמש בטלפון סלולארי

אין להשתמש במחשב אישי או

אין להשתמש בדיסק און קי ו/או מכשיר מדיה אחר

> אין להפריד את דפי שאלון -הבחינה

<u>בחינת סמסטר:</u> קיץ

<u>השנה: 2015</u>

<u>מועד:</u>

תאריך הבחינה: 210, 2015 שעת הבחינה: משך הבחינה: 180 בחינה:

מס' נבחן

<u>השאלון ייבדק בתום הבחינה</u> <u>ע"י המרצה</u>

מרצים:

<u>מבנה הבחינה והנחיות לפתרון:</u>

· MOJ SAILLE & SIAR MIND 2 - XITED SAILE VILLE 525 Mes 61 20 25 1 ve 4 mole, pour - mins retire 3

## בהצלחה!

כל הזכויות שמורות ©. מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכאנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה

לפניך 25 שאלות אמריקאיות. לכל שאלה תשובה **אחת** שהיא הנכונה ו∖או המדוייקת ביותר – בחר בה. שתי נקודות ינתנו עבור כל תשובה נכונה

|          |     | ערכת המוגדרת כ- soft real time ניתן לצפות ל interrupt latency ארוך יותר מאשר<br>ערכת המוגדרת hard real time. |
|----------|-----|--|
|          | א.  | ECI  |
|          | ב.  | לא נכון  |
|          | .ג  | להיפך  |
|          | ۲.  | תלוי   |
|          |     | ערכת הפועלת ללא מערכת הפעלה, קל יותר לתעדף משימות מסויימות מכיוון שהקוד<br>ר יותר.                           |
|          | א.  | לרב נכון   |
|          | ב.  | jitter תלוי ב  |
|          | ג.  | לרב לא נכון  |
|          | т.  | תמיד נכון  |
| :<br>:20 | אם  | בעל אינטרוול של 5mSec בעל אינטרוול של system tick במערכת קיים  |
|          | א.  | ה latency של המערכת גדול מ5mSec  |
|          | ב.  | 5mSec במהירות גבוהה יותר מאשר פעם ב-scheduling לא יתכן   |
|          | . ג | 5mSec -במהירות גבוהה יותר מאשר פעם ב context switch לא יתכן  |
|          | т.  | תשובות א', ב', ג' אינן נכונות  |
| 2        | כל  | watchdog- חייב לרוץ לפחות פעם בדקה כדי שלא יתקבל task  |
|          | א.  | נכון   |
|          | ב.  | task לא כל   |
|          | ג.  | תלוי אם זה מנגנון חומרה או תוכנה   |
|          | .Τ  | interrupt וגם כל task וגם כל   |
|          |     |  |

| scheduler-מתזמן את פעולת ה-system tick   | .5 |
|--|----|
| א. אמת צרופה   |    |
| ב. שטויות  |    |
| ג. תלוי  |    |
| ready-ים שהם במצב ready-ים שהם במצב ready  |    |
| ראם ייתכן שקטע קוד מסויים ירוץ ב-context-של context-ים שונים?  | .6 |
| א. לעולם לא  |    |
| ב. רק אם אין מערכת הפעלה   |    |
| ג. זה נכון עבור כל הקוד של האפליקציה   |    |
| re-entrant-ד. רק אם הוא הקוד הוא חלק מפונקציית   |    |
| מחסנית "יושבת" ב-ROM.  | .7 |
| א. תמיד  |    |
| ב. לפעמים  |    |
| ג. לעולם לא  |    |
| ד. רק אם יש מספיק מקום בזכרון  |    |
| האם ניתן לחסום פסיקות לפרק זמן קטן יותר מזה של הsystem-tick?   | .8 |
| א. לא ניתן   |    |
| ב. רק מתוך task-ים בעדיפות גבוהה מאד   |    |
| ג. ניתן  |    |
| ד. רק אם המעבד חזק מספיק   |    |
| ידוע שבמערכת מסויימת ה-idle task רץ לעיתים לפרק זמן רצוף של mSec30 , ניתן להסיק<br>שה interval של ה-system tick גדול\שווה 30mSec | .9 |
| א. אמת ויציב   |    |
| ב. לא  |    |
| ג. רק במערכת בה קיימים מספר מועט של task-ים  |    |
| ד. תלוי (במזג האויר למשל)  |    |
|  |    |

| software watchdog, אז ניתן לאמר software watchdog. אם במערכת קיים                                    |
|--|
| dead lock א. שהמערכת מוגנת בפני  |
| ב. שהמערכת מוגנת בפני priority inversion   |
| starvation ג. שהמערכת מוגנת בפני   |
| ד. תשובות א' ו ג' נכונות   |
| 11. במערכת ובה ממומש התקן watch dog חומרתי, פעולת ה strobe מבוצעת מה-context של ה<br>task בעדיפות ה- |
| א. גבוהה ביותר   |
| ב. לא משנה, העיקר שיינתן strobe  |
| ג. מכל ה-task-ים   |
| ד. נמוכה ביותר   |
| Mutex .12 משמש ל   |
| א. הגנה על גישה למשתנה גלובלי משותף  |
| ב. הגנה על משאב חומרה משותף  |
| ג. לצורך סנכרון  |
| ד. כל התשובות  |
| priority <b>inheritance</b> מונע בעיה של Mutex .13   |
| א. נכון  |
| ב. לא נכון   |
| ג. תלוי משאב עליו מגינים   |
| task של ה priority של ה  |
| mutex של ה priority של ה Task .14  |
| mutex-א. אם הוא מחזיק ב  |
| ב. אם הוא עשוי לבקש את ה-mutex   |
| ג. רק לאחר ששחרר את ה-mutex  |
|  |
|  |

| T        | mutex-ו bask -ו mutex נוסף מבקש את הtask ו mutex  |
|----------|---|
|          |   |
|          | ם דרושה לנו השהיה של mSec 8 ושל 16mSec באפליקציה, ובהנחה שניתן להשתמש רק ב<br>system call של מערכת ההפעלה בכדי לממש זאת, מה היית בוחר\ת לשמש כ interval של ה<br>system tic? |
| ٧.       | 2mSec   |
| 1        | 4mSec   |
| <b>ג</b> | 8mSec   |
| Ţ        | 16mSec  |
|          | מערכת מבוצע ה- ISR של ה-system-tick בכל 10mSec. מה מהבאים נכון, כאשר מבקשים<br>שהיה למשך שני "טיקים"? (ה-task המבקש הוא בפריוריטי <b>הנמוך</b> ביותר)                       |
| 4        | עשויה להתקבל השהיה ישל 30mSec   |
| 1        | 9mSec עשויה להתקבל השהיה של   |
| ג        | בהכרח תתקבל השהיה של 15mSec או יותר   |
| Т        | תמיד תתקבל השהיה של 25mSec לפחות  |
| 17. ל    | מה זקוקים למנגנון ה system tick במערכת?   |
| ĸ        | לצורך שרותי השהייה  |
| ב        | deadlock למניעת   |
| λ        | כדי לאפשר לכל ה task-ים לרוץ  |
| Ţ        | תשובות א' וב'   |
| 18. ב    | "טיקים taskDelay למשך 3 "טיקים task -אע ש- task   |
| א        | יתבצע context switch אחרי 3 טיקים   |
| ב        | יתבצע context switch ראופו מיידי  |

ג. יתבצע context switch באופן מיידי וגם לאחר 3 תיקים

ד. זה לא קשור ל context switch.

| שתנים לוקאליים של re-entrant function, נמצאים ב:   | 19. מ  |
|--|--------|
| ROM .  | א.     |
| . עשויים להמצא ב stack של כל הtaskים שקוראים לפונקציה זו.  | ב      |
| ב stack של הtask האחרון שקרא לפונקציה זו stack ב.  | ג.     |
| stack משתנים לוקאליים כלל לא נמצאים ב  | .Т     |
| פונקצייה שאיננה re-entrant עושים שימוש במשתנה גלובלי.  | 20. ב  |
| . חייבים להגן עליו באמצעות סמפור   | א      |
| חייבים להגן עליו באמצעות מיוטקס .  | ב      |
| אסור להגן עליו בשום מקרה.  | .λ     |
| . אין בהכרח חובה להגן עליו   | т      |
| קוד הנקרא מתוך context של task מסויים, יש שימוש במשאב משותף מסוג מערך גלובלי.<br>ה היא הדרך היעילה ביותר , מהיבט מהירות הביצוע, להגן עליו? |        |
| Mutex .  | א.     |
| . סמפור  | ב.     |
| Interrupt disable/enable   | ג.     |
| Scheduler lock   | .Τ     |
| תונה פונקציה ללא קריאות ל system calls כלשהן, ונתון שהפונקציה התחילה לרוץ כעת:   | 22. נר |
| במהלך ריצת הפונקציה לפחות עד סיומה context switch במהלך ריצת הפונקציה לפחות עד   | .א     |
| system call שקרא לפונקציה זו ימשיך לרוץ עד אשר יקרא ל task שקרא לפונקציה זו ימשיך לרוץ עד אשר יקרא ל                                       | ב.     |
| לא יתכן שה- scheduler ירוץ לפני סיום ריצת הפונקציה   | ג.     |
| scheduler כל פסיקה במהלך ריצת הפונקציה תגרום לריצת ה   | Τ,     |
| : waiting אמצב running דמצב Tas  | k .23  |
| SemaphorePend בכל פעם שתקרא הפונקציה   | א.     |
| interrupt במידה ויקרה  | ב.     |
| בכל קריאה ל TimeDelay בכל קריאה ל  | . ۲    |
|  |        |

| waiting-לא יכול לעבור מ- running לא יכול לעבור מ- Task  |
|---|
| ב: TCB .24  |
| RAM .א  |
| ROM .a  |
| ג. תלוי   |
| חסn volatile memory ד. ב  |
| Priority .25 נקבע על פי   |
| א. דחיפות הטיפול בארוע מרגע התרחשותו  |
| ב. גודל המחסנית של ה-task   |
| ג. מספר שורות הקוד שמכיל הtask  |
| taska של הlatency ד. ה latency של ה   |
| שאלות "פתוחות": בחר וענה על שתיים מתוך השלוש. 25 נקודות עבור כל שאלה  |
| Mutex .1  |
| א. למה הוא משמש?  |
| ב. איזו בעיה הוא פותר וכיצד?  |
| . תאר את אופן פעולתו בעזרת תרשים ריצה של task-ים על ציר הזמן.   |
| ד. מתי נכון לעשות בו שימוש ומתי אין צורך להשתמש בו?   |
| ה. מנה את ההבדלים בינו ובין סמפור.  |
| 2. תאר את מנגנון מנגנון ה-watch dog המשלב פתרון <b>תוכנתי וחומרתי</b> כפי שלמדנו בכיתה. (אין<br>המדובר על הפתרון החומרת <sup>,</sup> בלבד!) |
| ציין את דרך הפעולה. מתי הוא מופעל? אילו מקרים הוא בא למנוע?   |
| 3. בעיית הפילוסופים   |
| א. תאר אותה   |
| ב. הצע שתי דרכים לפתרון   |
| בהצלחה !!!  |

## 

DEPT. OF SYSTEMS ENGINEERING GRADUATE STUDIES

הפות'קה להם סת ותעונות Dubban - mil

בחינות

<u>שם הקורס:</u> מערכות הפעלה <u>קוד הקורס:</u> 10303

<u>הוראות לנבחן:</u>

חומר עזר שימושי לבחינה חומר סגור

אין לכתוב בעפרון

אין להשתמש בטלפון סלולארי -

אין להשתמש במחשב אישי או -

אין להשתמש בדיסק און קי ו/או מכשיר מדיה אחר

אין להפריד את דפי שאלון הבחינה

<u>מרצים:</u>

<u>בחינת סמסטר:</u> קיץ

<u>השנה:</u> 2015

תאריך הבחינה:

<u>שעת הבחינה:</u>

משך הבחינה:

<u>ע"י המרצה</u>

<u>השאלון ייבדק בתום הבחינה</u>

<u>מועד:</u>

מבנה הבחינה והנחיות לפתרון:

## בהצלחה!

כל הזכויות שמורות ©. מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן מאגר מידע, בכל דרך שהיא, בין מכאנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה

לפניך 25 שאלות אמריקאיות. לכל שאלה תשובה **אחת** שהיא הנכונה ו\או המדוייקת ביותר – בחר בה. שתי נקודות ינתנו עבור כל תשובה נכונה

|     | נרכת המוגדרת כ- soft real time ניתן לצפות ל interrupt latency ארוך יותר מאשר<br>ערכת המוגדרת hard real time. |
|-----|--|
| א.  | בכון   |
| .ع  | לא נכון  |
| . ג | להיפך  |
| ,Τ  | תלוי   |
|     | ערכת הפועלת ללא מערכת הפעלה, קל יותר לתעדף משימות מסויימות מכיוון שהקוד<br>ר יותר.                           |
| א.  | לרב נכון   |
| ב.  | jitter תלוי ב  |
| ג.  | לרב לא נכון  |
| ٦.  | תמיד נכון  |
| אם  | בעל אינטרוול של 5mSec בעל אינטרוול של system tick במערכת קיים  |
| .א  | ה latency של המערכת גדול מ5mSec  |
| ב.  | 5mSec במהירות גבוהה יותר מאשר פעם ב-scheduling לא יתכן   |
| ړ.  | 5mSec -במהירות גבוהה יותר מאשר פעם ב context switch לא יתכן  |
| .T  | תשובות א', ב', ג' אינן נכונות  |
| כל  | watchdog- חייב לרוץ לפחות פעם בדקה כדי שלא יתקבל task  |
| .א  | נכון   |
| ב.  | task לא כל   |
| ג.  | תלוי אם זה מנגנון חומרה או תוכנה   |
| т.  | interrupt וגם כל task וגם כל   |
|     |  |
|     | במ<br>ג. ב. א. מהמ<br>ג. ב. א. מהמ<br>ג. ב. א.   |

| scheduler- מתזמן את פעולת ה-system tick  | ัก | .5 |
|--|----|----|
| אמת צרופה  | א  |    |
| שטויות.  | ב  |    |
| . תלוי   | ג. |    |
| ready-ים שהם במצב task רק עבור .   | Τ. |    |
| אם ייתכן שקטע קוד מסויים ירוץ ב-context של taskים שונים?   | ה  | .6 |
| . לעולם לא   | א  |    |
| . רק אם אין מערכת הפעלה  | ב. |    |
| . זה נכון עבור כל הקוד של האפליקציה  | ג. |    |
| re-entrant-רק אם הוא הקוד הוא חלק מפונקציית.   | Τ. |    |
| מחסנית "יושבת" ב-ROM.  | מ  | .7 |
| . תמיד   | א  |    |
| לפעמים   | ב  |    |
| . לעולם לא   | ג. |    |
| . רק אם יש מספיק מקום בזכרון   | Т  |    |
| אם ניתן לחסום פסיקות לפרק זמן קטן יותר מזה של הsystem-tick?  | ה  | .8 |
| לא ניתן.   | א  |    |
| :. רק מתוך task-ים בעדיפות גבוהה מאד   | ב  |    |
| . ביתן   | ג. |    |
| רק אם המעבד חזק מספיק.   | T  |    |
| רוע שבמערכת מסויימת ה-idle task רץ לעיתים לפרק זמן רצוף של mSec30 , ניתן להסיק<br>אה interval של ה- system tick גדול\שווה 30mSec |    | .9 |
| ו. אמת ויציב   | א  |    |
| לא   | ב  |    |
| רק במערכת בה קיימים מספר מועט של task-ים.  | ג. |    |
| ר. תלוי (במזג האויר למשל)  | Т  |    |
|  |    |    |

.

| software watchdog, אז ניתן לאמר software watchdog   | 10. אם      |
|---|-------------|
| dead lock שהמערכת מוגנת בפני  | .א          |
| priority inversion שהמערכת מוגנת בפני   | .2          |
| starvation שהמערכת מוגנת בפני   | .λ          |
| תשובות א' ו ג' נכונות   | .т          |
| ערכת ובה ממומש התקן watch dog חומרתי, פעולת ה strobe מבוצעת מה-context של ה<br>t בעדיפות ה- |             |
| גבוהה ביותר   | .א          |
| לא משנה, העיקר שיינתן strobe  | ב.          |
| מכל ה-task-ים   | .λ          |
| נמוכה ביותר   | т.          |
| Mut משמש ל  | tex .12     |
| הגנה על גישה למשתנה גלובלי משותף  | .א          |
| הגנה על משאב חומרה משותף  | <b>.</b> 2  |
| לצורך סנכרון  | κ.          |
| כל התשובות  |             |
| priority <b>inheritance</b> מונע בעיה של Mut  | tex .13     |
| CI  | .א          |
| לא נכון   | .2          |
| תלוי משאב עליו מגינים   | .λ          |
| task של ה priority של ה   | .1          |
| mutex של ה priority של ה Ta   | isk .14     |
| mutex-אם הוא מחזיק ב  | .א          |
| mutex-אם הוא עשוי לבקש את ה   | ב.          |
| mutex-רק לאחר ששחרר את ה-mutex  | <b>.</b> د. |
| אם הוא מחזיק ב task, ו- task נוסף מבקש את ה-mutex   |             |
|   |             |

| דרושה לנו השהיה של mSec 8 ושל 16mSec באפליקציה, ובהנחה שניתן להשתמש רק ב system ca של מערכת ההפעלה בכדי לממש זאת, מה היית בוחר\ת לשמש כ interval של ה system te? | alls    |
|--|---------|
| 2mSec  | א.      |
| 4mSec  | ב.      |
| 8mSec  | ړ.      |
| 16mSec   | Τ.      |
| ערכת מבוצע ה- ISR של ה-system-tick בכל 10mSec. מה מהבאים נכון, כאשר מבקשים<br>יהיה למשך שני "טיקים"? (ה-task המבקש הוא בפריוריטי <b>הנמוך</b> ביותר)             |         |
| עשויה להתקבל השהיה של 30mSec   | א.      |
| עשויה להתקבל השהיה של 9mSec  | ב.      |
| בהכרח תתקבל השהיה של 15mSec או יותר  | .ג      |
| תמיד תתקבל השהיה של 25mSec לפחות   | ٦٦.     |
| ה זקוקים למנגנון ה system tick במערכת?   | 17. למו |
| לצורך שרותי השהייה   | .א      |
| deadlock למניעת  | ב.      |
| כדי לאפשר לכל ה task-ים לרוץ   | .ג      |
| 'תשובות א' וב  | т.      |
| "טיקים taskDelay למשך 3 "טיקים task מבצע task  | 18. ברו |
| יתבצע context switch אחרי 3 טיקים  | .א      |
| יתבצע context switch באופן מיידי   | ב.      |
| יתבצע context switch באופן מיידי וגם לאחר 3 תיקים  | .λ      |
| .context switch זה לא קשור ל   | .1      |
|  |         |

| יתנים לוקאליים של re-entrant function, נמצאים ב:  | 19. מש  | è  |
|---|---------|----|
| ROM   | א.      |    |
| עשויים להמצא ב stack של כל הtask-ים שקוראים לפונקציה זו.  | ב.      |    |
| ב stack של הtask האחרון שקרא לפונקציה זו  |         |    |
| stack משתנים לוקאליים כלל לא נמצאים ב   | Τ.      |    |
| ונקצייה שאיננה re-entrant עושים שימוש במשתנה גלובלי.  | 20. בפו | Ž. |
| חייבים להגן עליו באמצעות סמפור  | .א      |    |
| חייבים להגן עליו באמצעות מיוטקס   | .1      |    |
| אסור להגן עליו בשום מקרה  | .ג      |    |
| אין בהכרח חובה להגן עליו  | .T      |    |
| ד הנקרא מתוך context של task מסויים, יש שימוש במשאב משותף מסוג מערך גלובלי.<br>היא הדרך היעילה ביותר, מהיבט מהירות הביצוע, להגן עליו? |         |    |
| Mutex   | .א      |    |
| סמפור   | ב.      |    |
| Interrupt disable/enable  | .م      |    |
| Scheduler lock  |         |    |
| נה פונקציה ללא קריאות ל system calls כלשהן, ונתון שהפונקציה התחילה לרוץ כעת:  | 22. נתו | 51 |
| לא יתכן context switch במהלך ריצת הפונקציה לפחות עד סיומה   | .א      |    |
| system call שקרא לפונקציה זו ימשיך לרוץ עד אשר יקרא ל task ה  | .ء      |    |
| לא יתכן שה- scheduler ירוץ לפני סיום ריצת הפונקציה  | .ג      |    |
| כל פסיקה במהלך ריצת הפונקציה תגרום לריצת ה scheduler  | .т      |    |
| : waiting למצב running למצב Tanning   | isk .23 | 8  |
| SemaphorePend בכל פעם שתקרא הפונקציה  | .א      |    |
| interrupt במידה ויקרה   | ב.      |    |
| TimeDelay בכל קריאה ל   | .ג      |    |
| waiting-לא יכול לעבור מ- running לא יכול לעבור מ- Task  | .T      |    |
|   |         |    |

= 28

| :ב מצא ב: TCB .24   |
|---|
| RAM .x  |
| ROM .a  |
| ג. תלוי   |
| non volatile memory ב. ד  |
| Task של Task נקבע על פי   |
| א. דחיפות הטיפול בארוע מרגע התרחשותו  |
| ב. גודל המחסנית של ה-task   |
| ג. מספר שורות הקוד שמכיל הtask  |
| taska של הlatency ד. ה latency של ה   |
|   |
| שאלות "פתוחות": בחר וענה על שתיים מתוך השלוש. 25 נקודות עבור כל שאלה  |
| Mutex .1  |
| א. למה הוא משמש?  |
| ב. איזו בעיה הוא פותר וכיצד?  |
| ג. תאר את אופן פעולתו בעזרת תרשים ריצה של task-ים על ציר הזמן.  |
| ד. מתי נכון לעשות בו שימוש ומתי אין צורך להשתמש בו?   |
| ה. מנה את ההבדלים בינו ובין סמפור.  |
| 2. תאר את מנגנון מנגנון ה-watch dog המשלב פתרון <b>תוכנתי וחומרתי</b> כפי שלמדנו בכיתה. (אין<br>המדובר על הפתרון החומרתי בלבד!) |
| ציין את דרך הפעולה. מתי הוא מופעל? אילו מקרים הוא בא למנוע?   |
| 3. בעיית הפילוסופים   |
| א. תאר אותה   |

בהצלחה !!!

ב. הצע שתי דרכים לפתרון