



מס' נבחן



**שם הקורס: מערכות הפעלה**  
**קוד הקורס: 10303**

**הוראות לנבחן:**

- חומר עזר שימושי לבחינה
- חומר סגור

**בחינת סמסטר: קיץ**

**השנה: 2015**

**מועד:**



**תאריך הבחינה:** 13.10.2015

**שעת הבחינה:** 17:00

**משך הבחינה:** 180 דקות

**השאלון ייבדק בתום הבחינה**

**ע"י המרצה**

**מרצים:**

עזריאל מאיר

- אין לכתוב בעפרון
- אין להשתמש בטלפון סלולארי
- אין להשתמש במחשב אישי או נייד
- אין להשתמש בדיסק און קי ו/או מכשיר מדיה אחר
- אין להפריד את דפי שאלון הבחינה

**מבנה הבחינה והנחיות לפתרון:**

25 שאלות אמריקאיות - 2 נקודות עבור כל תשובה נכונה.  
נא לממן את התשובה הנכונה בכל השאלון.

3 שאלות במונח - מממן, יש לענות על שתיים, 25 נק' כל תשובה.

**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 ולא המדוייקת ביותר – בחר בה. שתי נקודות ינתנו עבור כל תשובה נכונה

1. במערכת המוגדרת כ- soft real time ניתן לצפות ל interrupt latency ארוך יותר מאשר במערכת המוגדרת hard real time.

א. נכון

ב. לא נכון

ג. להיפך

ד. תלוי

2. במערכת הפועלת ללא מערכת הפעלה, קל יותר לתעדף משימות מסויימות מכיוון שהקוד רץ מהר יותר.

א. לרב נכון

ב. תלוי ב jitter

ג. לרב לא נכון

ד. תמיד נכון

3. אם במערכת קיים system tick בעל אינטרוול של 5mSec ניתן לאמר ש

א. ה latency של המערכת גדול מ5mSec

ב. לא יתכן scheduling במהירות גבוהה יותר מאשר פעם ב- 5mSec

ג. לא יתכן context switch במהירות גבוהה יותר מאשר פעם ב- 5mSec

ד. תשובות א', ב', ג' אינן נכונות

4. כל task חייב לרוץ לפחות פעם בדקה כדי שלא יתקבל reset מה watchdog

א. נכון

ב. לא כל task

ג. תלוי אם זה מנגנון חומרה או תוכנה

ד. כל task וגם כל interrupt



5. ה- system tick מתזמן את פעולת ה-scheduler

א. אמת צרופה

ב. שטויות

ג. תלוי

ד. רק עבור task-ים שהם במצב ready

6. האם ייתכן שקטע קוד מסויים ירוץ ב-context של task-ים שונים?

א. לעולם לא

ב. רק אם אין מערכת הפעלה

ג. זה נכון עבור כל הקוד של האפליקציה

ד. רק אם הוא הקוד הוא חלק מפונקציית-re-entrant

7. מחסנית "יושבת" ב-ROM.

א. תמיד

ב. לפעמים

ג. לעולם לא

ד. רק אם יש מספיק מקום בזכרון

8. האם ניתן לחסום פסיקות לפרק זמן קטן יותר מזה של ה-system-tick?

א. לא ניתן

ב. רק מתוך task-ים בעדיפות גבוהה מאד

ג. ניתן

ד. רק אם המעבד חזק מספיק

9. ידוע שבמערכת מסויימת ה-idle task רץ לעיתים לפרק זמן רצוף של 30mSec, ניתן להסיק שה interval של ה-system tick גדול/שווה 30mSec

א. אמת ויציב

ב. לא

ג. רק במערכת בה קיימים מספר מועט של task-ים

ד. תלוי (במזג האויר למשל...)



10. אם במערכת קיים software watchdog, אז ניתן לאמר

- א. שהמערכת מוגנת בפני dead lock
- ב. שהמערכת מוגנת בפני priority inversion
- ג. שהמערכת מוגנת בפני starvation
- ד. תשובות א' ו ג' נכונות

11. במערכת ובה ממומש התקן watch dog חומרתי, פעולת ה strobe מבוצעת מה-context של ה task בעדיפות ה-

- א. גבוהה ביותר
- ב. לא משנה, העיקר שיינתן strobe
- ג. מכל ה-task-ים
- ד. נמוכה ביותר

12. Mutex משמש ל

- א. הגנה על גישה למשתנה גלובלי משותף
- ב. הגנה על משאב חומרה משותף
- ג. לצורך סנכרון
- ד. כל התשובות

13. Mutex מונע בעיה של priority inheritance

- א. נכון
- ב. לא נכון
- ג. תלוי משאב עליו מגינים
- ד. תלוי ב priority של ה task

14. Task מסוים עשוי לרוץ ב priority של ה mutex

- א. אם הוא מחזיק ב-mutex
- ב. אם הוא עשוי לבקש את ה-mutex
- ג. רק לאחר ששחרר את ה-mutex



ד. אם הוא מחזיק ב mutex ו- task נוסף מבקש את ה- mutex

15. אם דרושה לנו השהיה של 8 mSec ושל 16mSec באפליקציה, ובהנחה שניתן להשתמש רק ב system calls של מערכת ההפעלה בכדי לממש זאת, מה היית בוחרת לשמש כ interval של ה system tick?

א. 2mSec

ב. 4mSec

ג. 8mSec

ד. 16mSec

16. במערכת מבוצע ה-ISR של ה- system-tick בכל 10mSec. מה מהבאים נכון, כאשר מבקשים השהיה למשך שני "טיקים"? (ה- task המבקש הוא בפריווריטי הנמוך ביותר)

א. עשויה להתקבל השהיה של 30mSec

ב. עשויה להתקבל השהיה של 9mSec

ג. בהכרח תתקבל השהיה של 15mSec או יותר

ד. תמיד תתקבל השהיה של 25mSec לפחות

17. למה זקוקים למנגנון ה system tick במערכת?

א. לצורך שרותי השהיה

ב. למניעת deadlock

ג. כדי לאפשר לכל ה- task-ים לרוץ

ד. תשובות א' וב'

18. ברגע ש- task מבצע taskDelay למשך 3 "טיקים"

א. יתבצע context switch אחרי 3 טיקים

ב. יתבצע context switch באופן מיידי

ג. יתבצע context switch באופן מיידי וגם לאחר 3 טיקים

ד. זה לא קשור ל context switch.



19. משתנים לוקאליים של re-entrant function, נמצאים ב:

א. ROM

ב. עשויים להמצא ב stack של כל task-ים שקוראים לפונקציה זו.

ג. ב stack של task האחרון שקרא לפונקציה זו

ד. משתנים לוקאליים כלל לא נמצאים ב stack

20. בפונקצייה שאיננה re-entrant עושים שימוש במשתנה גלובלי.

א. חייבים להגן עליו באמצעות סמפור

ב. חייבים להגן עליו באמצעות מיוטקס

ג. אסור להגן עליו בשום מקרה

ד. אין בהכרח חובה להגן עליו

21. בקוד הנקרא מתוך context של task מסוים, יש שימוש במשאב משותף מסוג מערך גלובלי. מה היא הדרך היעילה ביותר, מהיבט מהירות הביצוע, להגן עליו?

א. Mutex

ב. סמפור

ג. Interrupt disable/enable

ד. Scheduler lock

22. נתונה פונקציה ללא קריאות ל system calls כלשהן, ונתון שהפונקציה התחילה לרוץ כעת:

א. לא יתכן context switch במהלך ריצת הפונקציה לפחות עד סיומה

ב. ה task שקרא לפונקציה זו ימשיך לרוץ עד אשר יקרא ל system call

ג. לא יתכן שה-scheduler ירוץ לפני סיום ריצת הפונקציה

ד. כל פסיקה במהלך ריצת הפונקציה תגרום לריצת ה scheduler

23. Task יעבור ממצב running למצב waiting :

א. בכל פעם שתקרא הפונקציה SemaphorePend

ב. במידה ויקרה interrupt

ג. בכל קריאה ל TimeDelay



ד. Task לא יכול לעבור מ- running ל- waiting

24. TCB נמצא ב:

א. RAM

ב. ROM

ג. תלוי

ד. non volatile memory

25. Priority של Task נקבע על פי

א. דחיפות הטיפול בארוע מרגע התרחשותו

ב. גודל המחסנית של ה-task

ג. מספר שורות הקוד שמכיל ה-task

ד. ה latency של ה-task

שאלות "פתוחות": בחר וענה על שתיים מתוך השלוש. 25 נקודות עבור כל שאלה

1. Mutex

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 תוכנתי וחומרתי כפי שלמדנו בכיתה. (אין המדובר על הפתרון החומרתי 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 !!!



פתרון



פתרון מוצג

13.10.2015



מס' נבחן

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

DEPT. OF SYSTEMS ENGINEERING  
GRADUATE STUDIES



המחלקה להנדסת מערכות  
לימודי תואר

בחינות

**שם הקורס: מערכות הפעלה**  
**קוד הקורס: 10303**

**בחינת סמסטר: קיץ**  
**השנה: 2015**  
**מועד:**

**הוראות לנבחן:**

- חומר עזר שימושי לבחינה
- חומר סגור

**תאריך הבחינה:**  
**שעת הבחינה:**  
**משך הבחינה:**

- אין לכתוב בעפרון
- אין להשתמש בטלפון סלולארי
- אין להשתמש במחשב אישי או נייד

**השאלון ייבדק בתום הבחינה**  
**ע"י המרצה**

- אין להשתמש בדיסק און קי ו/או מכשיר מדיה אחר
- אין להפריד את דפי שאלון הבחינה

**מרצים:**

**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 המדויקת ביותר – בחר בה. שתי נקודות ינתנו עבור כל תשובה נכונה

1. במערכת המוגדרת כ- soft real time ניתן לצפות ל interrupt latency ארוך יותר מאשר במערכת המוגדרת hard real time.

א. נכון

ב. לא נכון

ג. להיפך

ד. תלוי

2. במערכת הפועלת ללא מערכת הפעלה, קל יותר לתעדף משימות מסויימות מכיוון שהקוד רץ מהר יותר.

א. לרב נכון

ב. תלוי ב jitter

ג. לרב לא נכון

ד. תמיד נכון

3. אם במערכת קיים system tick בעל אינטרוול של 5mSec ניתן לאמר ש

א. ה latency של המערכת גדול מ5mSec

ב. לא יתכן scheduling במהירות גבוהה יותר מאשר פעם ב- 5mSec

ג. לא יתכן context switch במהירות גבוהה יותר מאשר פעם ב- 5mSec

ד. תשובות א', ב', ג' אינן נכונות

4. כל task חייב לרוץ לפחות פעם בדקה כדי שלא יתקבל reset מה watchdog

א. נכון

ב. לא כל task

ג. תלוי אם זה מנגנון חומרה או תוכנה

ד. כל task וגם כל interrupt



5. ה- system tick מתזמן את פעולת ה-scheduler

א. אמת צרופה

ב. שטויות

ג. תלוי

ד. רק עבור task-ים שהם במצב ready

6. האם ייתכן שקטע קוד מסויים ירוץ ב-context של task-ים שונים?

א. לעולם לא

ב. רק אם אין מערכת הפעלה

ג. זה נכון עבור כל הקוד של האפליקציה

ד. רק אם הוא הקוד הוא חלק מפונקציית-re-entrant

7. מחסנית "יושבת" ב-ROM.

א. תמיד

ב. לפעמים

ג. לעולם לא

ד. רק אם יש מספיק מקום בזכרון

8. האם ניתן לחסום פסיקות לפרק זמן קטן יותר מזה של ה-system-tick?

א. לא ניתן

ב. רק מתוך task-ים בעדיפות גבוהה מאד

ג. ניתן

ד. רק אם המעבד חזק מספיק

9. ידוע שבמערכת מסויימת ה-idle task רץ לעיתים לפרק זמן רצוף של 30mSec, ניתן להסיק שה interval של ה-system tick גדול/שווה 30mSec

א. אמת ויציב

ב. לא

ג. רק במערכת בה קיימים מספר מועט של task-ים

ד. תלוי (במזג האויר למשל...)



10. אם במערכת קיים software watchdog, אז ניתן לאמר

א. שהמערכת מוגנת בפני dead lock

ב. שהמערכת מוגנת בפני priority inversion

ג. שהמערכת מוגנת בפני starvation

ד. תשובות א' ו ג' נכונות

11. במערכת ובה ממומש התקן watch dog חומרתי, פעולת ה strobe מבוצעת מה-context של ה task בעדיפות ה-

א. גבוהה ביותר

ב. לא משנה, העיקר שיינתן strobe

ג. מכל ה-task-ים

ד. נמוכה ביותר

12. Mutex משמש ל

א. הגנה על גישה למשתנה גלובלי משותף

ב. הגנה על משאב חומרה משותף

ג. לצורך סנכרון

ד. כל התשובות

13. Mutex מונע בעיה של priority inheritance

א. נכון

ב. לא נכון

ג. תלוי משאב עליו מגינים

ד. תלוי ב priority של ה task

14. Task מסויים עשוי לרוץ ב priority של ה mutex

א. אם הוא מחזיק ב-mutex

ב. אם הוא עשוי לבקש את ה-mutex

ג. רק לאחר ששחרר את ה-mutex

ד. אם הוא מחזיק ב mutex, ו- task נוסף מבקש את ה-mutex



15. אם דרושה לנו השהיה של 8 mSec ושל 16mSec באפליקציה, ובהנחה שניתן להשתמש רק ב system calls של מערכת ההפעלה בכדי לממש זאת, מה היית בוחרת לשמש כ interval של ה system tick?

א. 2mSec

ב. 4mSec

ג. 8mSec

ד. 16mSec

16. במערכת מבוצע ה-ISR של ה-system-tick בכל 10mSec. מה מהבאים נכון, כאשר מבקשים השהיה למשך שני "טיקים"? (ה-task המבקש הוא בפריוריטי הנמוך ביותר)

א. עשויה להתקבל השהיה של 30mSec

ב. עשויה להתקבל השהיה של 9mSec

ג. בהכרח תתקבל השהיה של 15mSec או יותר

ד. תמיד תתקבל השהיה של 25mSec לפחות

17. למה זקוקים למנגנון ה system tick במערכת?

א. לצורך שרותי השהיה

ב. למניעת deadlock

ג. כדי לאפשר לכל ה task-ים לרוץ

ד. תשובות א' וב'

18. ברגע ש-task מבצע taskDelay למשך 3 "טיקים"

א. יתבצע context switch אחרי 3 טיקים

ב. יתבצע context switch באופן מיידי

ג. יתבצע context switch באופן מיידי וגם לאחר 3 טיקים

ד. זה לא קשור ל context switch.



19. משתנים לוקאליים של re-entrant function, נמצאים ב:

א. ROM

ב. עשויים להמצא ב stack של כל task-ים שקוראים לפונקציה זו.

ג. ב stack של task האחרון שקרא לפונקציה זו

ד. משתנים לוקאליים כלל לא נמצאים ב stack

20. בפונקצייה שאיננה re-entrant עושים שימוש במשתנה גלובלי.

א. חייבים להגן עליו באמצעות סמפור

ב. חייבים להגן עליו באמצעות מיוטקס

ג. אסור להגן עליו בשום מקרה

ד. אין בהכרח חובה להגן עליו

21. בקוד הנקרא מתוך context של task מסוים, יש שימוש במשאב משותף מסוג מערך גלובלי. מה היא הדרך היעילה ביותר, מהיבט מהירות הביצוע, להגן עליו?

א. Mutex

ב. סמפור

ג. Interrupt disable/enable

ד. Scheduler lock

22. נתונה פונקציה ללא קריאות ל system calls כלשהן, ונתון שהפונקציה התחילה לרוץ כעת:

א. לא יתכן context switch במהלך ריצת הפונקציה לפחות עד סיומה

ב. ה task שקרא לפונקציה זו ימשיך לרוץ עד אשר יקרא ל system call

ג. לא יתכן שה-scheduler ירוץ לפני סיום ריצת הפונקציה

ד. כל פסיקה במהלך ריצת הפונקציה תגרום לריצת ה scheduler

23. Task יעבור ממצב running למצב waiting :

א. בכל פעם שתקרא הפונקציה SemaphorePend

ב. במידה ויקרה interrupt

ג. בכל קריאה ל TimeDelay

ד. Task לא יכול לעבור מ- running ל- waiting



24. TCB נמצא ב:

א. RAM

ב. ROM

ג. תלוי

ד. non volatile memory

25. Priority של Task נקבע על פי

א. דחיפות הטיפול בארוע מרגע התרחשותו

ב. גודל המחסנית של ה-task

ג. מספר שורות הקוד שמכיל ה-task

ד. ה latency של ה-task

שאלות "פתוחות": בחר וענה על שתיים מתוך השלוש. 25 נקודות עבור כל שאלה

1. Mutex

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 תוכנתי וחומרתי כפי שלמדנו בכיתה. (אין המדובר על הפתרון החומרתי 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 !!!