

מס' נבחן

בחינות – המחלקה להנדסת תוכנה

שם הקורס: מערכות הפעלה
קוד הקורס: 10303

הוראות לנבחן:	בחינת סמסטר:
- חומר עזר שימושי לבחינה	השנה: 2016
חומר סגור	מועד: א'
- אין לכתוב בעפרון	תאריך הבחינה: 9.10.2016
- אין להשתמש בטלפון סלולארי	שעת הבחינה: 17:00
- אין להשתמש במחשב אישי או נייד	משך הבחינה: 3 שעות
- אין להשתמש בדיסק און קי ו/או מכשיר מדיה אחר	השאלון ייבדק בתום הבחינה
- אין להפריד את דפי שאלון הבחינה	ע"י המרצה
	מרצה: עדי מלאך

מבנה הבחינה והנחיות לפתרון:

50% (10 שאלות אמריקאיות 5 נקודות עבור כל תשובה נכונה. רק תשובה אחת נכונה לכל שאלה אמריקאית.
 50% (3 שאלות פתוחות, מתוכן יש לענות על שתיים בלבד – שאלה אחת היא חובה, ותוכלו לבחור אחת מתוך השתיים הנותרות.

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 10 שאלות אמריקאיות. לכל שאלה תשובה אחת שהיא הנכונה ולא המדויקת ביותר – בחר בה.
כל תשובה נכונה תזכה ב-5 נקודות.

1. שימוש ב mutex מפעיל מנגנון של priority inheritance
א. בכל פעם ש task מחזיק ב mutex
ב. לא
ג. תמיד
ד. תלוי
2. במידה ומתבצעת גישה למשאב משותף מתוך שני task ומתוך interrupt, שלושתם פונים לאותו משאב. נכון יהיה להגן על הפניה בעזרת שימוש ב scheduler lock
א. כן
ב. לא
ג. תלוי במקרה
ד. רק אם הגישה למשאב המשותף היא קצרה
3. שינוי priority של task בזמן ריצה, ל priority גבוהה יותר, שקול לקריאה של אותו task ל scheduler lock
א. לא
ב. כן
ג. תלוי ב priority
ד. תלוי ב state של task
4. קריאה ל OS_Sem_Accept() גורמת ל task הקורא להכנס למצב של waiting
א. כאשר ה semaphore count שווה ל - 0
ב. תמיד
ג. לא
ד. כאשר ה semaphore count גדול מ - 0
5. אחד מהשדות ב TCB הוא ה priority של ה task שרץ בכל רגע נתון.
א. נכון
ב. לא נכון
ג. תלוי
ד. נכון רק עבור ה idle task
6. ה scheduler עושה שימוש ב lookup table כדי
א. לחסוך מקום בזכרון
ב. לחסוך בזמן ריצה
ג. לשמור על מבנה מודולרי
ד. לדעת איזה task במצב ready

7. מתי מתבצע context switch בין task?

- א. בכל קריאה ל system call
- ב. בכל יציאה מ ISR
- ג. בכל קריאה ל OSTaskDelay
- ד. אף תשובה איננה נכונה

8. הקוד למימוש mutex מבצע במהלכו, בין היתר, פעולה של interrupt disable/enable.

- א. תלוי במימוש של האפליקציה
- ב. רק אם מתבצע priority inheritance
- ג. כן
- ד. לא

9. משך זמן ריצת ה scheduler במערכת ההפעלה שלמדנו מושפע מ:

- א. מספר ה task הפעילים במערכת
- ב. מספר ה task אשר במצב delayed
- ג. מהירות המעבד
- ד. ה priority של ה task שרץ כרגע

10. נתון שבמערכת מסוימת מתבצע קטע קוד 'אטומי' למשך זמן של 10mSec והוא חלק מפונקציה reentrant.

- א. יש להוסיף הגנה על קטע הקוד הזה מפני גישה משותפת
- ב. ה latency של המערכת יהיה בהכרח קטן מ 10mSec
- ג. בקטע הקוד האטומי אסור לעשות שימוש במשתנים גלובליים
- ד. בקטע הקוד האטומי אסור לקרוא ל system calls

שאלות 'פתוחות': ענה על שאלה 3 – היא שאלת חובה. ובחר וענה על אחת משתי השאלות 1 ו 2 . 25 נקודות עבור תשובה נכונה ומלאה.

1. (שאלת בחירה)

- א. 5% מהו starvation? תאר מה הם הגורמים להיווצרות מצב זה.
 - ב. 5% איך ניתן לזהות מצב זה במערכת סטנדרטית כפי שלמדנו בכיתה?
 - ג. 5% איך ניתן להמנע ממצב זה, ואיך ניתן לפתור מצב זה במידה וזוהה?
 - ד. 10% תאר מנגנון לזיהוי starvation במערכת ההפעלה שלמדנו, אך הפעם המערכת תדפיס למסך במידה ותזהה מצב זה, ולא תבצע reset.
- לתורך כך כתוב קוד של ה taskים המעורבים וציין איך היית מתכנן את המנגנון כולל priorities, ושאר שיקולים רלבנטיים.

2. (שאלת בחירה)

- א. במערכת מוגדר system tick interval של 5mSec. במערכת קיימים 3 taskים. task בעל priority הבינוני מבקש delay ל 3 system ticks. תאר בעזרת תרשים על ציר הזמן את מצבי הקצה בהם task ישוב לרוץ-באיזה תרחיש ישוב לרוץ מהר מאד, ובאיזה תרחיש הוא עשוי להמתין זמן רב?
- ב. מהו deadlock, מה גורם לו? כיצד ניתן למנוע אותו? תאר מנגנון לאיתור deadlock כפי למדנו בכיתה.

3. (שאלת חובה) עבור הקוד הבא, תאר את ההתרחשות על ציר הזמן מרגע הקריאה ל OSStart().

ולמשך 150mSec. כמו כן הראה מה יודפס על המסך. הנה שיצירת ה taskים וה semaphore בוצעו באופן תקין, ושההדפסות לא לוקחות זמן משמעותי.

כמו כן, במערכת מוגדר system tick interval של 10mSec.

א. כאשר ה priorities הן:

Task 1: priority 1
Task 2: priority 2
Task 3: priority 3
Task 4: priority 4

ב. כאשר ה priorities הן:

Task 1: priority 4
Task 2: priority 3
Task 3: priority 2
Task 4: priority 1

```
Task1 (void *p_arg)
{
    while (DEF_ON)
    {
        OSTimeDlyHMSM(0, 0, 0, 50);
        OSSemPost(App_Sem1);
        OSSemPost(App_Sem1);
        APP_TRACE_DBG("1");
    }
}
```

```
Task2(void *p_arg)
{
    while (DEF_ON)
```

```

    {
        OSSemPend(App_Sem1, 0, &err);
        APP_TRACE_DBG("2");
    }
}

Task3(void *p_arg)
{
    while (DEF_ON)
    {
        OSSemPend(App_Sem1, 0, &err);
        APP_TRACE_DBG("3");
    }
}

Task4(void *p_arg)
{
    while (DEF_ON)
    {
        OSSemAccept(App_Sem1);
        APP_TRACE_DBG("4");
        OSTimeDlyHMSM(0, 0, 20);
    }
}

```


פתרון



מס' נבחן

בחינות – המחלקה להנדסת תוכנה

שם הקורס: מערכות הפעלה
קוד הקורס: 10303

הוראות לנבחן:	בחינת סמסטר:
- חומר עזר שימושי לבחינה	השנה: 2016
חומר סגור	מועד: 7c
- אין לכתוב בעפרון	תאריך הבחינה: 9.10.16
- אין להשתמש בטלפון סלולארי	שעת הבחינה:
- אין להשתמש במחשב אישי או נייד	משך הבחינה:
- אין להשתמש בדיסק און קי ו/או שאלון "בדק בתום הבחינה	ע"י המרצה
- משיר מדיה אחר	מרצה: עדי מלאך
- אין להפריד את דפי שאלון הבחינה	

מבנה הבחינה והנחיות לפתרון:

- 50% (10 שאלות אמריקאיות 5 נקודות עבור כל תשובה נכונה. רק תשובה אחת נכונה לכל שאלה אמריקאית.
- 50% (3 שאלות פתוחות, מתוכן יש לענות על שתיים בלבד – ■■■ לה אחת היא חובה, ותוכלו לבחור אחת מתוך השתיים הנותרות .

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 10 שאלות אמריקאיות. לכל שאלה תשובה אחת שהיא הנכונה ו\או המדויקת ביותר – בחר בה.
כל תשובה נכונה תזכה ב-5 נקודות .

1. שימוש ב mutex מפעיל מנגנון של priority inheritance
א. בכל פעם ש task מחזיק ב mutex
ב. לא
ג. תמיד
ד. תלוי
2. במידה ומתבצעת גישה למשאב משותף מתוך שני task ומתוך interrupt, שלושתם פונים לאותו משאב. נכון יהיה להגן על הפניה בעזרת שימוש ב scheduler lock
א. כן
ב. לא
ג. תלוי במקרה
ד. רק אם הגישה למשאב המשותף היא קצרה
3. שינוי priority של task בזמן ריצה, ל priority גבוהה יותר, שקול לקריאה של אותו task ל scheduler lock
א. לא
ב. כן
ג. תלוי ב priority
ד. תלוי ב state של task
4. קריאה ל OS_Sem_Accept() גורמת ל task הקורא להכנס למצב של waiting
א. כאשר ה semaphore count שווה ל - 0
ב. תמיד
ג. לא
ד. כאשר ה semaphore count גדול מ - 0
5. אחד מהשדות ב TCB הוא ה priority של ה task שרץ בכל רגע נתון.
א. נכון
ב. לא נכון
ג. תלוי
ד. נכון רק עבור ה idle task
6. ה scheduler עושה שימוש ב lookup table כדי
א. לחסוך מקום בזכרון
ב. לחסוך בזמן ריצה
ג. לשמור על מבנה מודולרי
ד. לדעת איזה task במצב ready

7. מתי מתבצע context switch בין task?

- א. בכל קריאה ל system call
- ב. בכל יציאה מ ISR
- ג. בכל קריאה ל OSTaskDelay
- ד. אף תשובה איננה נכונה

8. הקוד למימוש mutex מבצע במהלכו, בין היתר, פעולה של interrupt disable/enable.

- א. תלוי במימוש של האפליקציה
- ב. רק אם מתבצע priority inheritance
- ג. כן
- ד. לא

9. משך זמן ריצת ה scheduler במערכת ההפעלה שלמדנו מושפע מ:

- א. מספר ה task הפעילים במערכת
- ב. מספר ה task אשר במצב delayed
- ג. מהירות המעבד
- ד. ה priority של ה task שרץ כרגע

10. נתון שבמערכת מסוימת מתבצע קטע קוד 'אטומי' למשך זמן של 10mSec והוא חלק מפונקציה reentrant.

- א. יש להוסיף הגנה על קטע הקוד הזה מפני גישה משותפת
- ב. ה latency של המערכת יהיה בהכרח קטן מ 10mSec
- ג. בקטע הקוד האטומי אסור לעשות שימוש במשתנים גלובליים
- ד. בקטע הקוד האטומי אסור לקרוא ל system calls

שאלות 'פתוחות': ענה על שאלה 3 – היא שאלת חובה. ובחר וענה על אחת משתי השאלות 1 ו 2 .
25 נקודות עבור תשובה נכונה ומלאה.

1. (שאלת בחירה)

- 5% מהו starvation? תאר מה הם הגורמים להיווצרות מצב זה.
 - 5% איך ניתן לזהות מצב זה במערכת סטנדרטית כפי שלמדנו בכיתה?
 - 5% איך ניתן להמנע ממצב זה, ואיך ניתן לפתור מצב זה במידה וזוהה?
 - 10% תאר מנגנון לזיהוי starvation במערכת ההפעלה שלמדנו, אך הפעם המערכת תדפיס למסך במידה ותזהה מצב זה, ולא תבצע reset.
- לתורך כך כתוב קוד של ה taskים המעורבים וציין איך היית מתכנן את המנגנון כולל priorities, ושאר שיקולים רלבנטיים.

2. (שאלת בחירה)

- במערכת מוגדר system tick interval של 5mSec. במערכת קיימים 3 taskים. task בעל priority הבינוני מבקש delay ל 3 system ticks. תאר בעזרת תרשים על ציר הזמן את מצבי הקצה בהם task ישוב לרוץ-באיזה תרחיש ישוב לרוץ מהר מאד, ובאיזה תרחיש הוא עשוי להמתין זמן רב?
- מהו deadlock, מה גורם לו? כיצד ניתן למנוע אותו? תאר מנגנון לאיתור deadlock כפי למדנו בכיתה.

3. (שאלת חובה) עבור הקוד הבא, תאר את ההתרחשות על ציר הזמן מרגע הקריאה ל OSStart().

ולמשך 150mSec. כמו כן הראה מה יודפס על המסך. הנה שיצירת ה taskים וה semaphore בוצעו באופן תקין, ושההדפסות לא לוקחות זמן משמעותי.

כמו כן, במערכת מוגדר system tick interval של 10mSec.

א. כאשר ה priorities הן:

Task 1: priority 1

Task 2: priority 2

Task 3: priority 3

Task 4: priority 4

ב. כאשר ה priorities הן:

Task 1: priority 4

Task 2: priority 3

Task 3: priority 2

Task 4: priority 1

```
Task1 (void *p_arg)
{
    while (DEF_ON)
    {
        OSTimeDlyHMSM(0, 0, 0, 50);
        OSSemPost(App_Sem1);
        OSSemPost(App_Sem1);
        APP_TRACE_DBG("1");
    }
}
```

```
Task2(void *p_arg)
{
    while (DEF_ON)
```



```

    {
        OSSemPend(App_Sem1, 0, &err);
        APP_TRACE_DBG("2");
    }
}

Task3(void *p_arg)
{
    while (DEF_ON)
    {
        OSSemPend(App_Sem1, 0, &err);
        APP_TRACE_DBG("3");
    }
}

Task4(void *p_arg)
{
    while (DEF_ON)
    {
        OSSemAccept(App_Sem1);
        APP_TRACE_DBG("4");
        OSTimeDlyHMSM(0, 0, 20);
    }
}

```