

# מערכות הפעלה 31261

מבחן סופי, מועד א', סמסטר ב' תשע"ג, 14/07/2013

**הוראות לנבחן:** משך הבחינה שעתיים וחצי. חומרי העזר המותרים הם השקפים של הקורס שנמסרו באתר הקורס ומחשבונים פשוטים בלבד. כל חומר נוסף מעבר למותר ייפסול את הבחינה! רשום את תשובותיך במחברת במצורפת לבחינה. ציין בבירור את מספר השאלה במחברת והשתדל להיות קצר וענייני (תשובות ארוכות מדי או נסיונות לתת מספר תשובות אפשריות ייפסלו את תשובתך). הקפד על כתב יד ברור ומסודר, ומחק את כל חומר הטייטא. השימוש בשפה האנגלית מותר. השאלון מכיל 10 שאלות בשווי של 100 נקודות ושאלת בונוס בשווי 10 נקודות. ונפרש על פני 3 עמודים.  
**בהצלחה!**

## שאלה 1 [10%]

- תאר בקצרה מהו תהליך חישוב במערכת מחשב?
- מהם המרכיבים השונים מהם הוא מורכב? פרט מהם ההבדלים העקריים בין data ובין Heap.
- מהם המצבים השונים בהם עשוי תהליך להיות? הסבר בקצרה את המאפיין העיקרי של כל מצב.

## שאלה 2 [10%]

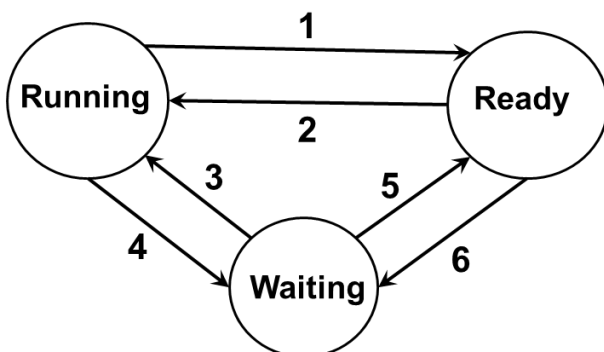
- תאר בקצרה מהו חוט (Thread) ומהם ההבדלים העקריים שבין חוט לתהליך?
- מנה את כל הדרכים שבהם חוטים שונים (באותו תהליך) יכולים לשתף מידע?
- מנה לפחות שלושה דרכים שבאמצעותם תהליכים שונים עשויים לשתף מידע

## שאלה 3 [5%]

- תאר בקצרה (ובאופן סכימטי בלבד) כיצד מערכת ההפעלה מבררת את המידע הבא עבור תהליך נתון
- מהי הכתובת ההתחלתית של הקוד שצריך להריץ?
  - מהי הכתובת הראשונה בזיכרון Heap פנוי?
  - הערך ההתחלתי של אוגר המחסנית (Stack Pointer)?

## שאלה 4 [18%]

לפניך דיאגרמת מצבים בהם עשוי להימצא תהליך הרץ במערכת הפעלה מרובת תהליכים. כל חץ מסמן מעבר ממצב אחד למצב שני. תאר בקצרה את התנאים שעשויים לגרום לתהליך לעבור ממצב אחד למצב שני (אם המעבר אפשרי) ורשום את תשובותיך בהתאם למספר החץ בדיאגרמה. במידת האפשר, תן דוגמאות לתנאים כאלה.



רשום את תשובתך במחברת הבחינה בפורמט הבא:

- חץ 1: תאור התנאים למעבר ...
- חץ 2: תאור התנאים למעבר ...
- חץ 3: תאור התנאים למעבר ...
- חץ 4: תאור התנאים למעבר ...
- חץ 5: תאור התנאים למעבר ...
- חץ 6: תאור התנאים למעבר ...

### שאלה 5 [5%]

בתהליך P במערכת ההפעלה Linux יש 10 חוטים (Threads) שונים. שישה חוטים מבליים 70% מזמנם בכתיבה לדיסק ו-30% מזמנם ביחידת העיבוד (CPU). ארבעת החוטים הנותרים מבליים 20% מזמנם באיסוף חבילות תיקשורת (מול כרטיס התיקשורת) ו-80% מהזמן בעיבוד CPU. בהנחה שמערכת ההפעלה שלנו היא הוגנת (Fair), מהו הזמן הכולל שבו התהליך P מבלה ביחידת העיבוד המרכזית?

### שאלה 6 [10%]

מה יתרחש לאחר שגריץ את התוכנית C הבאה במערכת ההפעלה Unix. פרט בקצרה את כל השלבים בדרך.

```
#include <unistd.h>
#include <stdio.h>
int main()
{
    int pid, pip[2] ;
    char instring[7] ;

    pipe(pip) ;

    pid = fork() ;
    if (pid == 0) {
        write(pip[1], "Hi Mom!", 7) ;
    } else {
        read(pip[0], instring, 7) ;
        printf("%s\n", instring) ;
    }
}
```

### שאלה 7 [10%]

רשום תוכנית קצרה בשפת C או בשפת Python שמתמשת בקריאות מערכת (כמו: fork, waitpid, exit, kill), ומבצעת את הדברים הבאים:

- תהליך אב מוליד תהליך בן ומדפיס את מספר התהליך שלו (pid)
- תהליך הבן מוליד תהליך נכד והולך לישון 120 שניות
- תהליך האב הולך לישון 60 שניות ואז מסיים את כל התהליכים ומסיים בעצמו

### שאלה 8 [15%]

נתונים לנו שני וקטורים u, v באורך 8000 כל אחד. המכפלה של שני וקטורים מוגדרת להיות סכום מכפלות האיברים המתאימים:  $u * v = \sum_{i=0}^n u[i] * v[i]$ , כאשר n מציין את אורך הוקטור (במקרה שלנו: n=8000). כתוב קוד של Python לביצוע המכפלה במקביל על ידי 8 תהליכים שונים. המאמץ החישובי חייב להתחלק במידה שווה בין כל 8 התהליכים.

**הכוונה:** הנח שווקטור הוא רשימה (list) או סדרה (tuple) פשוטה ב-Python. הנח שברשותך מעבד מרובה ליבות (ככל שיידרשו). מדובר בתהליכים (Processes) ולא בחוטים (Threads) לכן אתה עשוי להזדקק לתור (Queue) אחד או יותר בכדי לשתף מידע בין התהליכים השונים.

## שאלה 9 [7%]

עייך היטב בתוכנית Python הבאה :

```
a = 2 ; b = 4 ; c = 6

def foo():
    global a, b, c
    b = 2*a + 1
    c = b - 1

def bar():
    global a, b, c
    a = 2*c - 1
    b = b + 1

t1 = Thread(target=foo) ; t2 = Thread(target=bar)
t1.start() ; t2.start()
t1.join() ; t2.join()
sum = a+b+c
print sum
```

מהם כל התוצאות האפשריות של הפלט (sum) של התוכנית? תן נימוק קצר לתשובתך.

## שאלה 10 [10%]

בפרוייקט תוכנה רפואית, יש שלושה חוטים t1, t2, t3, של תהליך מסוים הניגשים לארבעה קבצים file1, file2, file3, file4, למטרות כתיבה בתדירות גבוהה מאוד. פונקציית הכתיבה שכל חוט עשוי להפעיל בכל זמן נתון היא פשוטה :

```
def lwrite(file,line):
    f = open(file, "a")
    f.write(line)
    f.close()
```

והיא נמשכת כמובן פרק זמן קצר מאוד. אך למרות זאת, מספר שבועות לאחר שיחרור המוצר, הלקוחות נתקלו במצבים בהם אותו הקובץ נפתח במקביל על ידי שני חוטים ותוצאת הכתיבה יצרה שיבושים (שמהווים סכנה לבריאות המטופלים). בכדי להתגבר על הבעיה, יש לתקן את הפונקציה lwrite() על ידי שימוש במנגנוני מניעה הדדית (Lock או Semaphore) בכדי להבטיח את התנאים הבאים :

- כל חוט יוכל במוקדם או במאוחר לגשת לכל קובץ שייבחר (מניעת מצב "הרעבה")
- בכל רגע נתון יהיה לכל היותר חוט אחד שכותב לקובץ נתון (כלומר : שני חוטים לא יוכלו לכתוב לאותו קובץ בו-זמנית)
- בכל רגע נתון לא יהיה ניתן לכתוב ליותר משני קבצים במקביל (כלומר : לא יכתבו שלושה קבצים במקביל, גם לא על ידי שלושה חוטים שונים!), אך לא תמנע כתיבה לשני קבצים על ידי שני חוטים. תקן את הפונקציה lwrite() בכדי שדרישות אלה ייתמלאו באופן סביר ולא בזבזני.

## שאלת בונוס [10%]

כתוב פונקציית Python בשם dirsize(dir) המקבלת ספרייה dir, ומחזירה את גודל הכולל של כל הקבצים הרגילים בספרייה (רקורסיבית).  
הכוונה: יש לעשות שימוש בפונקציות ספרייה: os.path.isfile, os.path.getsize, os.path.listdir, os.path.isdir  
אין להשתמש בפונקציה os.walk!