

מערכות הפעלה 31261

מבחן סופי, מועד ב', סמסטר א' תשע"ד, 25/02/2014

הוראות לנבחן: משך הבחינה שלוש שעות. חומרי העזר המותרים הם השקפים של הקורס שנמסרו באתר הקורס בלבד. שימוש במחשבוניו ומכשירים אלקטרוניים כלשהם אסור בהחלט. 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 11 שאלות בשווי של 106 נקודות, ונפרש על פני 3 עמודים.
בהצלחה!

שאלה 1 [12%]

- א. תאר בקצרה את מנגנון מצבי המשתמש והגרעין (user mode, kernel mode) הקיים בכל יחידת עיבוד מודרנית, וכיצד מערכת ההפעלה משתמשת בו על מנת להגן על משאבי המערכת מפני נזקים אפשריים (מצד תהליכי משתמש רגילים)?
- ב. תן לפחות שתי דוגמאות שונות של פעולות משתמש שכיחות, בהם מערכת ההפעלה מגנה על משאבי המערכת באמצעות מנגנון זה.

שאלה 2 [12%]

- תאר שלושה תפקידים **מרכזיים** של מערכת ההפעלה הקשורים לניהול **מערכת הזיכרון (RAM)** של מערכת המחשב, ותאר **בדיוק שלושה** סוגי בעיות שונות שמערכת ההפעלה צריכה להתמודד איתם במסגרת זו.

שאלה 3 [10%]

- א. הסבר את מושג פונקציות **API** בהקשר של תוכנות מערכות הפעלה
- ב. רשום שלוש דוגמאות של פונקציות **API** במערכת הפעלה **Linux**
- ג. מנה שתי סיבות טובות להעדפת שימוש בפונקציות **API** בתוכנות מערכות הפעלה על פני שימוש ישיר בקריאות מערכת (system calls)

שאלה 4 [10%]

- תאר לפחות שלושה ייתרונות שונים לכך שמבנה הנתונים Pipe שמשמש לתיקשורת בין תהליכים שונים מנוהל על ידי הגרעין (kernel) של מערכת ההפעלה, ושהחוצץ שלו (buffer) שמור בשטח הזיכרון של מערכת ההפעלה (system memory) ולא בשטח הזיכרון של תהליך משתמש (user memory).

שאלה 5 [8%]

- א. תאר בקצרה את מנגנון ההרשאות (permissions) עבור קריאה, כתיבה, והפעלה של **קבצים** במערכת ההפעלה Unix (התייחס לקבצים רגילים בלבד, לא לתיקיות!).
- ב. למשתמש dany יש ספרייה בשם tasks בחשבון ה-Unix שלו: /home/dany/tasks. רשום תוכנית בשפת Python בכדי לפתוח את כל הקבצים בתיקיה tasks לכתיבה וקריאה עבור כל המשתמשים השייכים לקבוצה שלו. על התוכנית לבצע רק את הפעולות המבוקשות ולא לגרום לתוצאות לוואי שלא התבקשו!

שאלה 6 [6%]

תאר במשפט אחד מה בדיוק מתבצע על ידי תוכנית Python הבאה:

```
import time, os
a = time.time()
b = ""
d = "D:/BRAUDE"
for path,dirs,files in os.walk(d):
    for f in files:
        if not f[-4:] == '.doc': continue
        p = os.path.join(path,f)
        x = os.path.getmtime(p)
        if x<a:
            a = x
            b = p
print b
```

הערה: הפונקציה `os.path.getmtime` מחזירה את זמן השינוי האחרון של קובץ (בשניות משנת 1970).

שאלה 7 [5% + 5% bonus]

רשום תוכנית קצרה שבה תהליך אב P מוליד תהליך בן C כך שמתקיימים הדברים הבאים:

- התהליך P מעביר את המשפט "Hello Son" לתהליך C, שמדפיס את המשפט הזה
- התהליך C מעביר את המשפט "Hello Father" לתהליך P אשר מדפיס גם הוא משפט זה
- סדר ההדפסה חייב להיות זהה למתואר בסעיפים הקודמים
- דרוש קיצור ושימוש מינימלי במשאבים!

הערה: תוכל להשתמש בשפת C או בשפת Python בכדי לרשום את התוכנית שלך

שאלת בonus: תוספת בonus של 5 נקודות יינתנו לכל מי שירשום תוכנית Python (או C) שתבצע את התסריט הנ"ל במלואו ובמדויק. (ניקוד יינתן לתשובה מדויקת בלבד! לא יינתן ניקוד חלקי עבור תשובה חלקית!)

שאלה 8 [10%]

הפלט של הפקודה ps של Linux (בתוספת הדגלים aux) נראה כך:

```
Linux> ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.4   5648 2540 ?        Ss   Apr20    0:01 /sbin/init showopts
avahi     360  0.0  0.2   3248 1072 ?        Ss   Apr20    0:00 avahi-daemon
root      363  0.0  0.1   5044   540 ?        Ss   Apr20    0:13 /usr/sbin/haveged -w 1024 -v 0 -F
root     2617  0.0  0.1   3884   564 ?        Ss   Apr20    0:00 /usr/bin/kdm
root     2700  0.0  0.2   5084  1180 ?        Ss   Apr20    0:03 /usr/lib/postfix/master
dina     3513  0.0  0.6   9196  3428 ?        S    Apr20    0:00 xterm
dany     2728  0.0  0.2   4528  1308 ?        Ss   Apr20    0:00 /bin/bash
samy     2794  0.0  0.2   7060  1088 ?        Sl   Apr20    0:00 /usr/bin/VBoxClient --clipboard
samy     3018  0.0  0.1   2496   740 ?        S    Apr20    0:00 python
samy     3491  0.0  0.5   5656  2792 pts/2    Ss   Apr20    0:01 bash
samy     9399 50.0  0.2   4476  1028 pts/2    R+   07:29    0:00 ps aux
```

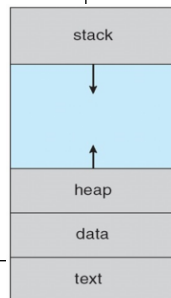
על בסיס צורת הפלט הזו, רשום תוכנית Python המשתמשת בפקודה `subprocess.Popen` לשם הפעלת התוכנית `ps aux`, אשר מדפיסה את רשימת המשתמשים שצורכים יותר מ-10% מהזיכרון במערכת (%MEM)

שאלה 9 [8%]

תוכנית C שלפניך היא חלק מקוד של תהליך P שרץ כרגע במערכת:

```
#define DAY 18
int year = 2014 ;
char *message = "Hello World" ;
double price = 517.23 ;

int funky(int a1, int a2) {
    char *pstr ;
    if(a1<a2) {
        long int y1=a1, y2=a2 ;
        char buffer[500] ;
        pstr = new char[150] ;
        return y1+y2 ;
    } else if (a2>17) {
        return a1 ;
    }
    return DAY+a1+a2 ;
}
```



- ערוך את רשימת כל המשתנים המופיעים בקוד זה
- עיין בקוד של התוכנית ונסה לשייך את כל המשתנים השונים שמופיעים שם לשטחי הזיכרון `heap`, `data`, או `stack` של התהליך P. רשום טבלה מתאימה שמציגה את תשובתך.
- ציין את הסכנה העיקרית שעשויה לנבוע כתוצאה מגידול בלתי מבוקר של שטחי ה-`heap` וה-`stack` ואיך היא נקראת בשפה מקצועית?

שאלה 10 [8%]

במערכת מחשב בעלת 8 ליבות ידוע כי לכל תהליך מנת זמן CPU מקסימלית של 40ms ולכל פסיקה להחלפת תהליכים (context switch) נדרש 3ms. בהנחה (אידיאלית) שכל התהליכים במערכת מנצלים את מלוא המנה המוקצית להם בכל ריצה, מהו המספר המקסימלי של תהליכים שהמערכת מסוגלת להריץ בשניה אחת? הערה: הנח שכל התהליכים התחילו לרוץ על המערכת לפני שעה כך שבשניה המדוברת שום תהליך חדש לא התחיל או סיים.

שאלה 11 [12%]

עיין היטב בתוכנית `producer.py` אשר פגשנו מספר פעמים במהלך הקורס.

```
import sys, time

def make_apples(n):
    for i in range(n):
        time.sleep(0.01)
        print "Apple_%d" % i
        sys.stdout.flush()

if __name__ == "__main__":
    n = int(sys.argv[1])
    make_apples(n)
```

- הסבר בשני משפטים מה התוכנית עושה וכיצד משתמשים בה מתוך שורת הפקודה של Unix (command line)?
- תוכנית המערכת `/usr/bin/gzip` היא תוכנית סטנדרטית במערכת ההפעלה Linux לשם דחיסת נתונים. בצרוף הדגל `-f` תוכנית זו מסוגלת לקלוט נתונים גם דרך הקלט הסטנדרטי (`stdin`), ולפלוט את הדחיסה שלהם לפלט הסטנדרטי (`stdout`). רשום תוכנית Python בשם `zip_apples.py` שמטרתה להפנות את הפלט הסטנדרטי של `producer.py` לקלט הסטנדרטי של התוכנית `gzip` לשם דחיסתו. לשם פשוטות, התוכנית `zip_apples.py` תדפיס את האורך של הנתונים לאחר דחיסתם.