האוניברסיטה הפתוחה

גיליון תשובות

מלא את הפרטים בכל המקומות הדרושים

מספר תעודת זהות (9 ספרות) | מספר הקורס | מספר השאלון

הדבק כאן את מדבקת הנבחן

סמסטר בחינה: _____ מועד: _____

תאריך בחינה: _____ מספר חדר: _____

אנא קרא את "הוראות חשובות לסטודנט" בגב דף זה.

לשימוש הבודקים

עבור בחינות פטור בלבד

☐ נכשל ☐ עבר

נא לסמן ב- X את המשבצת מעל הציון הקובע, לאחר בדיקה שנייה.

בדיקה שנייה

שם הבודק	ציון בחינה
	ציון שאלה 1
	ציון שאלה 2
	ציון שאלה 3
	ציון שאלה 4
	ציון שאלה 5
	ציון שאלה 6
	ציון שאלה 7
	ציון שאלה 8
	ציון שאלה 9
	ציון שאלה 10
	ציון שאלה 11
	ציון שאלה 12
	ציון שאלה 13
	ציון שאלה 14
	ציון שאלה 15
	ציון שאלה 16
	ציון שאלה 17
	ציון שאלה 18
	ציון שאלה 19
	ציון שאלה 20
	ציון שאלה 21
	ציון שאלה 22
	ציון שאלה 23
	ציון שאלה 24

בדיקה ראשונה

שם הבודק	ציון בחינה
	ציון שאלה 1
	ציון שאלה 2
	ציון שאלה 3
	ציון שאלה 4
	ציון שאלה 5
	ציון שאלה 6
	ציון שאלה 7
	ציון שאלה 8
	ציון שאלה 9
	ציון שאלה 10
	ציון שאלה 11
	ציון שאלה 12
	ציון שאלה 13
	ציון שאלה 14
	ציון שאלה 15
	ציון שאלה 16
	ציון שאלה 17
	ציון שאלה 18
	ציון שאלה 19
	ציון שאלה 20
	ציון שאלה 21
	ציון שאלה 22
	ציון שאלה 23
	ציון שאלה 24

E105 - 0457

מספר התלמיד הנבחן
רשום את כל תשע הספרות

הדבק כאן את
מדבקת הנבחן

האוניברסיטה
הפתוחה

ו' בתמוז תשע"ו

12
ביולי 2016

מס' שאלון - 534

סמסטר 2016ב

מס' מועד 84

4 / 20594

שאלון בחינת גמר

20594 - מערכות הפעלה

משך בחינה: 3 שעות

בשאלון זה **12** עמודים

מבנה הבחינה:

קראו בעיון לפני שתתחילו בפתרון הבחינה!

א. המבחן מורכב משלושה חלקים.

ב. בחלקים א ו - ב מופיעות שאלות פתוחות. ענו תשובות מלאות, בכתב קריא ובקיצור נמרץ. אין חובה להשתמש בכל השורות המוקצות לצורך התשובות, אך אין לחרוג מהמקום המוקצה.

ג. בחלק ג (שאלות אמריקאיות) עליכם לבחור בכל פעם בתשובה יחידה מבין התשובות המוצעות ולהקיף בעיגול את אות התשובה שבחרתם.

חומר עזר:
כל חומר עזר אסור בשימוש, פרט למחשבון, שאינו אוצר מידע.

בהצלחה !!!

החזירו
למשגיח את השאלון
וכל עזר אחר שקיבלתם בתוך מחברת התשובות

חלק א (55 נקודות)

ענו על **שלוש** השאלות הבאות.

שאלה 1 (24 נקודות)

ממשו סמפור (semaphore) לתיאום בין מספר תהליכונים באותו תהליך ע״י שימוש ב-pipe.
אפשר להגביל את המימוש שלכם על ידי כך שלא יתמוך במונה של סמפור אשר ערכו גדול מ- PIPE_BUF.

תזכורת:

Function: int **pipe** (*int filedes*`[2]`)

The `pipe` function creates a pipe and puts the file descriptors for the reading and writing ends of the pipe (respectively) into *filedes*`[0]` and *filedes*`[1]`.

If successful, `pipe` returns a value of `0`. On failure, `-1` is returned.

Reading or writing pipe data is *atomic* if the size of data written is not greater than `PIPE_BUF`. This means that the data transfer seems to be an instantaneous unit, in that nothing else in the system can observe a state in which it is partially complete. Atomic I/O may not begin right away (it may need to wait for buffer space or for data), but once it does begin it finishes immediately.

Reading or writing a larger amount of data may not be atomic; for example, output data from other processes sharing the descriptor may be interspersed. Also, once PIPE_BUF characters have been written, further writes will block until some characters are read.

```
typedef struct Semaphore_t {
        int ds[2];
} Semaphore;

int sem_init(Semaphore* sem, int count) {    //assume pointer is allocated
        ____________________________________________________
        ____________________________________________________
        ____________________________________________________
}


int sem_wait(Semaphore* sem) {
        ____________________________________________________
        ____________________________________________________
        ____________________________________________________
}


int sem_post(Semaphore* sem) {
        ____________________________________________________
        ____________________________________________________
}
```

הערה: אין צורך לכתוב קוד המטפל בכישלון של קריאות מערכת וגם אפשר להניח שהתוכנית שמשתמשת בסמפורים מתעלמת מ SIGPIPE.

שאלה 2 (16 נקודות)

הוכיחו או הפריכו את הטענות הבאות לגבי אלגוריתם הבנקאי (banker algorithm):

(5 נק') א. אם אלגוריתם הבנקאי גילה שהמערכת נמצאת במצב לא בטוח, אז בהכרח שהמערכת תגיע לקיפאון בעתיד.

(5 נק') ב. אם אלגוריתם הבנקאי גילה שהמערכת נמצאת במצב בטוח, אנחנו יודעים בוודאות שלא יתכן deadlock בעתיד.

(6 נק') ג. אם אלגוריתם הבנקאי גילה שהמערכת נמצאת במצב בטוח, אנחנו יודעים בוודאות שנוכל לסיים את הריצה בלי להגיע ל-deadlock.

שאלה 3 (15 נקודות)

במערכת מקבילית ישנם T תהליכונים (threads), כאשר T הוא מספר קבוע קטן מ 1024. במערת מספר מעבדים. בכל עת התהליכונים יכולים להיות מועברים למעבד אחר (לא חשוב כעת לפי איזה קריטריון). הבעיה היא שהתהליכונים יכולים "ליפול" בזמן לא ידוע ומסיבה לא ידועה. כאשר תהליכון "נופל" הוא מפסיק לבצע את הוראות שלו ולא חוזר לעולם.

תהליכונים יכולים לבצע פעולות שדורשות מתהליכונים אחרים להמתין. הפעולות הללו מוגבלות בזמן וחסומים ע"י הזמן שלוקח לבצע את הקוד הבא:

```
int counter;
for (i=0; i++; i<MAX_INT) {
        counter++;
}
```

נרצה לנצל את העובדה הזאת כדי לזהות את נפילת תהליכונים במערכת המקבילית, הרי אם תהליכוכשלהו מחזיק במנעול לאורך הזמן שעולה על זמן הביצוע של הלולאה דלעיל – הרי שזה אומר שתהליכון "נפל".

הוצעו 2 הצעות לפתור את הבעיה בעזרת מנעול אשר נפתח בעצמו אם הוא תפוס יותר מדי זמן.

ההצעה הראשונה הייתה כדלהלן:

```
typedef struct Lock_t
{
        bool    b;
} Lock;
```

```
void down(Lock* lock) {
    int counter = 0;
    while ( !t&s( &( lock →b) ) ) {
        counter++;
        if (counter == MAX_INT) {
            up(lock);
        }
    } // end of while
}
```

```
void up(Lock* lock) {
    bzero(lock→b);
}
```

ההצעה השניה הייתה:

```
typedef struct Lock_t
{
        bool      b;
        int       counter;
} Lock;
```

```
void down(Lock* lock) {
     int oldCounter;
     while ( !t&s( &(lock→b) ) ) {
          oldCounter =
             fetch&Increment( &(lock→counter) );
          if ( (oldCounter +1)  == MAX_INT) {
                up(lock);
          }
     } // end of while
     lock→ counter = 0;
}
```

```
void up(Lock* lock)
{
    lock→b = 0;
    lock→counter = 0;
}
```

הבהרות:

- תניחו ש-t&s() היא פקודה אטומית אשר קוראת ערך מכתובת ואם הוא אפס משנה אותו באופן אטומי ל-1. הפקודה מחזירה true אם הצליחה לשנות ערך. אחרת היא מחזירה false.

- תניחו ש-fetch&Increment() היא פקודה אטומית אשר קוראת ערך מכתובת, מקדמת את הערך שלו ב-1 ואחרי כך כותבת אותו בחזרה וכל זה באופן אטומי. הפקודה מחזירה את הערך הישן שנקרא (לפני עדכון).

(7 נק') א. הסבירו מדוע ההצעה הראשונה יכולה לגרום לכך שלא תתקיים מניעה הדדית (mutual exclusion):

(8 נק') ב. הוכיחו או הפריחו האם ההצעה שניה גם כן סובלת מבעיה של חוסר מניעה הדדית. הסבירו מדוע:

_____

_____

_____

חלק ב (25 נקודות)

ענו על השאלות הבאות. משקל כל שאלה 5 נקודות.

שאלה 4

האם הפונקציה fork יכולה להיכשל? אם לא, הסברו מדוע. אם כן, תנו דוגמא למצב כישלון אפשרי.

_____

_____

שאלה 5

מערכת העפלה לינוקס מחזיקה טבלת תהליכים (process table) בזיכרון ראשי. כל כניסה (entry) בטבלת תהליכים מכילה מידע על סיגנלים אשר נשלחו לתהליך. מדוע מידע זה נכלל בטבלת התהליכים וכתוצאה מכך נמצא כל הזמן בזיכרון ראשי? הסבירו איזו בעיה הייתה מתאוררת היה והיינו שומרים מידע זה יחד עם דפים רגילים של תהליכים.

_____

_____

שאלה 6

הסבירו מה מתבצע בתוכנית הבאה:

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

int
main(void)
{
      int n, fd[2];
      pid_t pid;
      char  line[2048];

      if (pipe(fd) < 0)
            fprintf(stderr,"pipe error");

      if ( (pid = fork()) < 0)
            fprintf(stderr,"fork error");

      else if (pid > 0) {                 /* parent */
            close(fd[0]);
```

```
            /* redirection of stdout so that stdout
                    of the process would be actualy
               written to the pipe
            */
                 close(1); /* stdout */
            dup(fd[1]);
            close(fd[1]);
            write(1, "hello world\n", 12);

      } else {                    /* child */
            close(fd[1]);

            /* redirection of stdin so that stdin would
               be actualy read from pipe
            */
            close(0); /*stdin*/
            dup(fd[0]);
            close(fd[0]);
            n = read(0, line, 2048);
            write(1, line, n);
      }

      exit(0);
}
```

שאלה 7

מדוע נועלים דפים בזיכרון בעת העברת נתונים על ידי DMA (Direct Memory Access)?

שאלה 9

מערכת הקבצים של מערכת הפעלה מסוימת משתמשת בשיטת ה I-node.

- גודל הבלוק במערכת הקבצים הוא 2 Kbyte
- כתובת הבלוק היא 8 בתים (bytes)
- 10 שדות של ה I-node יכולים להחזיק ישירות כתובת הבלוק בדיסק שלושה שדות נוספים:
- שדה הנועד להחזיק את הכתובת של ה single indirect block
- שדה הנועד להחזיק את הכתובת של ה double indirect block
- שדה הנועד להחזיק את הכתובת של ה triple indirect block

גודלו של קובץ מסוים במערכת הוא 632 Kbyte. חשבו מהי כמות הבלוקים שדרושה להחזקת קובץ זה במערכת הקבצים ותנו הסבר מפורט לחישוב.

חלק ג (20 נקודות)

שאלות רב-ברירה (אמריקאיות). משקל כל שאלה 5 נקודות.

שאלה 10

במערכת ניהול זיכרון מדפדף נהוגה מדיניות של prepaging – הבאת מספר כלשהו של דפים השייכים לתהליך בכל פעם שהתהליך עובר מהדיסק לזיכרון. בחרו טענה נכונה:

א. הבאת קבוצת דפים שנבחרה בקפידה היא פעולה שיכולה להוריד את כמות פסיקות הדפים (page faults) במערכת

ב. אין טעם להביא דפים מראש שכן הדבר כרוך בפעולת פינוי בשלב מאוחר יותר

ג. מדיניות זו עומדת בסתירה לעקרון קבוצת העבודה

ד. מדיניות זו ניתנת למימוש רק בשילוב עם האלגוריתם האופטימאלי להחלפת הדפים

שאלה 11

בחרו את הפעולה היקרה ביותר במונחים של מעברי בלוקים של הדיסק (disk block transfers) בהנחה שלא קיימים נתונים רלוונטיים בזיכרון המטמון (buffer cache):

א. פתיחת קובץ באמצעות open

ב. קריאת בלוק אחד באמצעות read

ג. קריאת תו אחד באמצעות getc

ד. התשובות א' וב' הן הנכונות

שאלה 12

איזו פעולה מן הפעולות הבאות אפשר לבצע אך ורק במצב ראשוני (kernel mode) במערכת ההפעלה Linux?

א. חסימת פסיקות החומרה (disabling hardware interrupts)

ב. החלפת תהליכונים (thread switch) כאשר מדובר בספריית תהליכונים ברמת המשתמש

ג. השמת ערך במשתנה גלובאלי

ד. את כל שלוש הפעולות הנ"ל יש לאפשר אך ורק במצב ראשוני

שאלה 13

לפניכם פסאודו-קוד של משחק רובוטים שבו 3 רובוטים בצבעים (אדום, כחול, ירוק) מבצעים תזוזות בסדר כלשהו.

```
int sem[3];
sem[0] = 1; sem[1] = 1; sem[2] = 1;
```

R_Robot (){	G_Robot (){	B_Robot (){
while(true) {	while(true) {	while(true) {
down(sem[0]);	down(sem[1]);	down(sem[2]);
<Make Move>	<Make Move>	<Make Move>
up(sem[1]);	up(sem[2]);	up(sem[0]);
}	}	}
}	}	}

```
int main(){
        run_new_thread(R_Robot);
        run_new_thread(G_Robot);
        run_new_thread(B_Robot);
}
```

האם סדר התזוזות של הרובוטים נקבע על ידי השימוש בסמפורים כדלהלן?
אם כן, מהו הסדר?

א. כן. אדום, ירוק, כחול וחוזר חלילה

ב. כן. אדום, כחול,ירוק וחוזר חלילה

ג. כן. אדום, כחול, אדום, כחול, ירוק וחוזר חלילה

ד. לא. הסדר ייקבע על ידי מתזמן (scheduler)

ב ה צ ל ח ה !