שאלה 1

במערכת נתונה רצים 4 תהליכים: A,B,C,D. במערכת זו ישנם 10 משאבים זהים מסוג R ואלו הם המשאבים היחידים הנדרשים על-ידי התהליכים.

לפניכם תיאור של שני מצבים שונים שבהם יכולה להימצא המערכת.

עליכם להחליט האם מצבים אלו הם מצבים הבטוחים מפני קיפאון (safe state) או מצבים בלתי בטוחים (unsafe state).

מצב א:

שם התהליך	מספר המשאבים התפוסים על-ידי תהליך זה	מספר המשאבים המקסימלי שהתהליך יכול לדרוש
A	1	6
B	1	5
C	2	4
D	4	7

מצב ב:

שם התהליך	מספר המשאבים התפוסים על-ידי תהליך זה	מספר המשאבים המקסימלי שהתהליך יכול לדרוש
A	1	6
B	2	5
C	2	4
D	4	7

בחרו בטענה הנכונה:

שני המצבים, א' וב', הם מצבים בטוחים.
שני המצבים, א' וב', הם מצבים בלתי בטוחים.
מצב א' הוא מצב בטוח ואילו מצב ב' הוא מצב בלתי בטוח.
מצב ב' הוא מצב בטוח ואילו מצב א' הוא מצב בלתי בטוח.

שאלה 2

תזמון תהליכים על פי שיטת Round Robin מבטיח כי :

לא יתקיים קפאון בין תהליכים

כמות התהליכים המסיימים ביחידת זמן תגדל בהשוואה לשיטת SJF

כמות התהליכים המסיימים ביחידת זמן תקטן בהשוואה לשיטת SJF

אף תשובה קודמת איננה בהכרח נכונה

שאלה 3

מערכת ההפעלה נתונה מספקת קריאת מערכת get_in(). להלן הפסיאודו-קוד שלה :

```
get_in( int * key ){
    int tmp;
    tmp = *key;
    *key = 1;
    return( tmp );
}
```

מערכת ההפעלה מבטיחה כי במהלך קריאת מערכת זו, לא תתבצע החלפת תהליכים.

לפתרון בעיית הקטע הקריטי במערכת זו, מוצע הפרוטוקול הבא :

```
int lock=0; /* Common variable, initialed to 0 at the beggining */

while( 1 )    /* Loop forever */
{
    /* Enter Critical Section */
    while ( get_in( &lock ) == 1 ); /* Wait until you can enter */

    Critical_Section();

    lock = 0;
}
```

האם זהו פתרון לבעיית הקטע הקריטי?

כן. אך רק לזוג תהליכים.

כן. לכל קבוצה של תהליכים.

לא. לא ניתן לפתור את בעיית הקטע הקריטי בעזרת משתנה יחיד.

לא. בעיית הקטע הקריטי ניתנת לפתרון בעזרת ארבעה משתנים לכל הפחות.

שאלה 4

אחד ההבדלים בין פונקציות ספרייה בשפת C (כגון ()printf) לבין קריאות מערכת (system calls) ב-UNIX הוא:

- רק למנהל המערכת (super user) מותר להשתמש בקריאות מערכת, בפונקציות ספרייה יכול להשתמש כל אחד.
- באמצעות פונקציות הספרייה בשפת C לא ניתן לפתוח קבצים לפעולות קריאה או כתיבה.
- כל קריאות המערכת הן חלק מפונקציות הספרייה של השפה.
- כל פונקציות הספרייה משתמשות בקריאות מערכת אך לא להפך.
- כל התשובות הקודמות אינן נכונות.

שאלה 5

כמה קריאות read יכולות להתבצע במקביל ע"י תהליך (process) אחד?

- לכל היותר קריאה אחת, שכן התהליך כולו נחסם לריצה עד לסיום הקריאה
- כל היותר x קריאות, כאשר x הוא מספר ה-user threads בתהליך
- לכל היותר x קריאות, כאשר x הוא מספר ה-kernel threads בתהליך
- לכל היותר x קריאות, כאשר x הוא מספר ה-threads מכל סוג שהוא בתהליך

שאלה 6

בחרו את הטענה הנכונה:

- מבנה פיקוח (monitor) חייב להיות מוגדר כמבנה של שפת תכנות (language structure)
- מבנה פיקוח (monitor) הוא שירות המסופק על-ידי מערכת הפעלה ולא חייב להיות מוגדר בשפת תכנות.
- שפת תכנות לבדה מסוגלת לספק סמפורים (semaphores) גם ללא תמיכתה של מערכת הפעלה המספקת שירותי סינכרוניזציה בין תהליכים.
- כל שלוש התשובות הקודמות הן נכונות

שאלה 7

בחנו את הקוד הבא למניעה הדדית בין N תהליכים. בחרו טענה נכונה לגבי הפרוטוקול הנ״ל:

```
process k  (1<=k <=N)
while (true) {
 for (i = 0; i < N; i++) {
    q[k] = i ;
    turn[i] = k ;
    L:  for (j = 0; j < N; j++) {
          if (j == k) continue;
          if ((q[j] >= i) && (turn[i] == k))
              goto L;
     }
 }
 q[k] = N;

 <critical section>

 q[k] = 0;
}
```

אחד מהתהליכים עלול להישאר תמיד מחוץ לקטע הקריטי ללא יכולת להיכנס

שני תהליכים יכולים להימצא בו-בזמן בתוך הקטע הקריטי

תהליך P1 לא יוכל להיכנס לקטע הקריטי בשום מקרה

הפרוטוקול הנ״ל מהווה פתרון לבעיות המרוץ (race condition problems) למרות שהוא משתמש בהמתנה פעילה (busy waiting)

שאלה 8

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

ביטול חסימת פסיקות החומרה (re-enabling hardware interrupts)

שינוי המצב של תהליכון (thread) מ-blocked ל-ready כאשר מדובר בספריית תהליכונים ברמת המשתמש.

ביצוע הפונקציה dup()

התשובות א׳ ו-ג׳ נכונות

התשובות ב׳ ו-ג׳ נכונות

שאלה 9

מצאו כמה רמות היררכיה דרושות לטבלת הדפים של תהליך אם:

- ✓ גודל דף הוא 4K Bytes
- ✓ מרחב הכתובות הוא בן 64 ביטים
- ✓ גודל של יחידת התייחסות הקטנה ביותר הוא בית אחד (1 byte)
- ✓ אין כל אפשרות להבטיח הקצאה רצופה של יותר מדף אחד עבור חלק כלשהו של הטבלה
- ✓ כל כניסה בטבלה היא בת 4 בתים (Bytes)

6

5

2

כל מספר בין 2-6 אפשרי בתנאים הנ״ל

שאלה 10

מערכת זמן אמת (real-time system) מסוגלת לתזמן את המשימות הבאות:

Period -תקופתיות של המשימה	Execution time- זמן ביצוע	Task
8	1	A
32	2	B
8	1	C
16	3	D
8	1	E

כמה משימות מסוג E ניתן לתזמן (בנוסף לאלה שכבר קיימות), כך שהמערכת תוכל עדיין לתזמן את כל המשימות? ניתן להניח כי זמן החלפת התהליכים (context switch) הוא זניח.

0

1

3

6

שאלה 11

תהליך כלשהו מבצע מתחילת ביצועו ועד סיומו סדרת גישות בזו אחר זו ל-P דפים בזיכרון. הסדרה מכילה התייחסויות ל-N דפים שונים (בסדר כלשהו).

מספר ה-frames לרשות התהליך הוא M. מהם החסמים על מספר ה-page faults האפשריים בכל מדיניות החלפת דפים שהיא?

לכל היותר M ולכל הפחות P-N

לכל היותר N ולכל הפחות P

לכל היותר P ולכל הפחות N-M

לכל היותר P ולכל הפחות N

שאלה 12

איזו מהטענות הבאות לגבי סיגנלים (signals) במערכת הפעלה UNIX נכונה :

סיגנלים הנשלחים לתהליכים נשלחים על-ידי גרעין של מערכת ההפעלה בלבד

סיגנלים נשלחים כתוצאה מקריאות מערכת (system calls) בלבד

סיגנלים נשלחים על-ידי מתאמי התקנים (device drivers) בלבד

אף תשובה קודמת איננה נכונה

שאלה 13

מי __בפועל__ מבצע שחרור משאבים והחזרתם למאגר המשאבים הפנויים של תהליך שהסתיים במערכת הפעלה Unix?

תהליך ההורה

גרעין המערכת

שד (daemon) מיוחד שמיועד לפעולות הנ"ל

אף תשובה קודמת איננה נכונה

שאלה 14

מניעת הקיפאון (deadlock) על-ידי תקיפת התנאי "מניעה הדדית" באמצעות שיתוף משאבים איננה ישימה בכל המקרים כי:

לא כל משאב ניתן לשיתוף

גורמת לכניסה למצב לא בטוח (unsafe state)

גורמת לכניסה למצב בטוח (safe state)

אף תשובה קודמת איננה נכונה

שאלה 15

מבנה פיקוח (monitor) מספק שתי פונקציות לעבודה עם condition variables: wait ו-signal. במימוש של מבנה פיקוח על פי Hoare הוצע שתהליך אשר קרא לפונקציה signal יושהה באופן מיידי ובמקומו ירוץ התהליך שזה עתה הוער. מדוע Hoare הציע מימוש זה? בחרו בתשובה הנכונה ביותר.

מימוש זה מונע היווצרות של מצבי דשדוש (thrashing) בעת שתהליכים מבצעים פרוצדורות של monitor

מימוש זה מונע היווצרות של מצבי קיפאון (deadlock) בין תהליכים שמשתמשים בפונקציות של monitor

מימוש זה מבטיח שרק תהליך אחד יימצא ב monitor

כל שלוש התשובות הנ"ל נכונות

שאלה 16

נתונה FAT עם קובץ A שמתחיל בבלוק 2, קובץ B שהבלוק הראשון שלו הוא 9 וקובץ C שמתחיל בבלוק 8. האם ה FAT עקבית (consistent)?

Block	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
Next block		15	12	11	18	-1	16		13	01	03	-1	06	04		10	5	-1	17

ה- FAT עקבית

ה- FAT איננה עקבית כי ל-A ול-B ישנו בלוק משותף

ה- FAT איננה עקבית כי ל-A ול-C ישנו בלוק משותף

ה- FAT איננה עקבית כי ל-B ול-C ישנו בלוק משותף

שאלה 17

באיזו שכבה של תוכנת קלט פלט ממומש לרוב buffer cache?

במערכת הטיפול בפסיקות הנוצרות על-ידי ההתקנים (interrupt handling mechanism)

בתוך תכניות התיאום (device drivers)

בתוך תוכנת קלט/פלט הבלתי תלויה בהתקן (device independent software)

בשדים (daemons) לארגון הפלט ובפונקציות הספרייה שאינן תלויות חומרה

שאלה 18

תהליך יחיד שנכנס ללולאה אינסופית נמצא במצב:

קיפאון (deadlock)

הרעבה (starvation)

דשדוש (thrashing)

אף תשובה קודמת איננה נכונה

שאלה 19

מתאמי התקנים (device drivers) במחשב יכולים להיות משני סוגים: מתאמי התקנים שמבצעים בדיקות בדבר סיום עבודת התקן ומתאמי התקנים שמבקשים מהתקנים לייצר interrupt בסיום עבודת ההתקן. מהו החיסרון המובהק של מתאמי התקנים מן הסוג הראשון בהשוואה למתאמי התקנים מן הסוג השני?

מתאמי התקנים מן הסוג הזה מסובכים למימוש

מתאמי התקנים מן הסוג הזה גורמים למספר גדול של פעולות I/O (במקרה של התקני I/O)

מתאמי התקנים מן הסוג הזה גורמים לבזבוז זמן CPU עקב ה- busy waiting

כל התשובות הקודמות הן נכונות

המשך הבחינה בעמוד הבא

שאלה 20

כמה תהליכים **חדשים** ייווצרו בעקבות ההרצה של התוכנית הבאה, כאשר ניתן להניח הצלחת כל קריאות המערכת?

```
main(){
        pid_1 = fork();
        pid_2 = fork();
        pid_3 = fork();
        pid_4 = fork();
}
```

2

7

15

17

בהצלחה!