

עקרונות ותכנון מערכות הפעלה ה'

הוראות לנבחנים

א. משך הבחינה: ארבע שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה ארבעה פרקים.

פרק ראשון	45 נקודות
פרק שני	18 נקודות
פרק שלישי	18 נקודות
פרק רביעי	19 נקודות
סך-הכול	100 נקודות

ג. חומר עזר מותר בשימוש: מחשבון פשוט שאינו ניתן לתכנות.

ד. הוראות מיוחדות:

- את התשובות לכל השאלות יש לכתוב **אך ורק** על גבי דף התשובות שבנספח א'.
- ענו על מספר השאלות הנדרש בשאלון. המעריך יעריך את מספר התשובות הנדרש בלבד, לפי סדר כתיבתן, ולא יתייחס לתשובות נוספות.
- לשאלון זה מצורף מילון מונחים בשפות עברית, ערבית, אנגלית ורוסית. תוכלו להיעזר בו בעת הצורך.

הוראות למשגיחים:

בתום הבחינה, יש לוודא שהנבחנים הדביקו את מדבקת הנבחן שלהם במקום המיועד לכך בדף התשובות שבנספח א', וצירפו אותו למחברת הבחינה.

כתבו **במחברת הבחינה בלבד**, בעמודים נפרדים, כל מה שברצונכם לכתוב **כטייטה** (ראשי פרקים, חישובים וכדומה).
כתבו "טייטה" בראש כל עמוד טייטה. כתיבת טייטות כלשהן על דפים שמחוץ למחברת הבחינה עלולה לגרום לפסילת הבחינה!

בשאלון זה 14 עמודים ו-2 עמודי נספחים.

השאלות בשאלון זה מנוסחות בלשון רבים,
אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

בהצלחה!

המשך מעבר לדף

השאלות

פרק ראשון (45 נקודות)

ענו על שאלה 1, הכוללת 15 סעיפים (א'–ט"ו). יש לענות על כל הסעיפים. הסעיפים א'–ט"ו אינם תלויים זה בזה, אלא אם כן מצוין אחרת. בכל הסעיפים נתונות ארבע תשובות, שרק אחת מהן נכונה. בכל סעיף בחרו את התשובה הנכונה, והקיפו בעיגול את הספרה המייצגת אותה בדף התשובות שבנספח א'.

שאלה 1 (45 נקודות – לכל סעיף 3 נקודות)

סעיפים א'–ג' מתייחסים לנתונים שלהלן:

במחשב מסוים הזיכרון מנוהל **כזיכרון מדומה** (virtual memory) הממומש בשיטת "**דפים נדרשים**" (demand paging).

גודלו של כל דף הוא 512 בתים.

להלן הצהרה של מטריצה D:

```
double D[128][128]
```

נתון כי טיפוס הנתונים double גודלו 8 בתים.

נתון כי המטריצה D שמורה בזיכרון בדפים.

המטריצה שמורה באופן הזה: האיבר הראשון D[0][0] שמור בתחילת הדף; איברי השורה הראשונה במטריצה שמורים לפני איברי השורה השנייה.

להלן קטע קוד:

```
for (int i = 0; i < 128; i++)  
    for (int j = 0; j < 128; j++)  
        D[i][j] = 0 ;
```

א. כמה החלפות דפים, **כולל הטעינה הראשונית** של המטריצה, יתרחשו במחשב אם הוקצו למטריצה **שש מסגרות בלבד** והמטריצה לא נמצאת בזיכרון בתחילת החישוב?

1. 128

2. 256

3. 512

4. 1,024

ב. כמה החלפות דפים, **כולל הטעינה הראשונית** של המטריצה, יתרחשו במחשב אם הוקצו למטריצה **עשר מסגרות בלבד** והמטריצה לא נמצאת בזיכרון בתחילת החישוב?

1. 128

2. 256

3. 512

4. 1,024

ג. להלן קטע קוד חדש, שמחליף את קטע הקוד הנתון בפתיח לסעיפים א'-ג':

```
for (int j = 0; j < 128; j++)
    for (int i = 0; i < 128; i++)
        D[i][j] = 0 ;
```

כמה החלפות דפים, **כולל הטעינה הראשונית** של המטריצה, יתרחשו **כעת** במחשב אם הוקצו למטריצה **חמש מסגרות בלבד** והמטריצה לא נמצאת בזיכרון בתחילת החישוב?

1. 128

2. 4,096

3. 16,384

4. 131,072

סעיפים ז'-ח' מתייחסים לנתונים שלהלן:

נתונה מערכת הפעלה שמנהלת את הזיכרון **כזיכרון מזומה** (virtual memory) הממומש בשיטת "**דפים נדרשים**" (demand paging).

אלגוריתם הקצאת הדפים הוא אלגוריתם סטטי, ואלגוריתם החלפת הדפים הוא בשיטת "ראשון מגיע, ראשון יוצא" (FIFO). נתון תהליך P הבנוי **משמונה** דפים, ועם יצירתו מוקצות לו **שלוש מסגרות בלבד** בזיכרון הממשי.

תהליך P דורש את הדפים לפי הסדר שלהלן (קראו משמאל לימין) –

1, 2, 3, 6, 2, 8, 4, 1, 6, 5, 2, 6

ז. איזה דף יימצא בזיכרון הממשי בתום ביצוע התהליך P, במסגרת **השנייה** שהוקצתה לו?

1. 1

2. 2

3. 5

4. 6

ה. מהו יחס ההחטאה (miss ratio), כלומר מהו אחוז כְּשלי הדף (page fault)?

1. 0.66

2. 0.75

3. 0.83

4. 1

ו. אם החלפת הדפים תתבצע בשיטת LRU, איזה דף יימצא בזיכרון הממשי בתום ביצוע התהליך P, במסגרת **השלישית** שהוקצתה לו?

1. 1

2. 2

3. 5

4. 6

ז. אם החלפת הדפים תתבצע בשיטת LRU, מה יהיה יחס ההחטאה (miss ratio), כלומר מהו אחוז כְּשלי הדף (page fault)?

1. 0.16

2. 0.33

3. 0.66

4. 0.83

ח. מה יהיה הדף הוותיק ביותר בזיכרון בסוף ריצת התהליך?

1. 1
2. 2
3. 6
4. 7

סעיפים ט'-י"א מתייחסים לנתונים שלהלן:

הטבלה שלהלן מתארת את תור ההמתנה לקבלת המעבד (ready queue) בנקודת זמן מסוימת t_0 .

כל חמשת התהליכים מוכנסים לתור בו זמנית בנקודת זמן t_0 .

הם מוכנסים לתור לפי הסדר הבא (מימין לשמאל): P_5, P_4, P_3, P_2, P_1 .

לצד כל תהליך מצוין ה-CPU burst time שלו.

כמו כן, בנקודת זמן t_0 , מתזמן המעבד (CPU scheduler) מקצה בפעם הראשונה את המעבד לאחד מחמשת התהליכים שבתור.

הניחו כי במהלך כל פרק הזמן שחולף בין t_0 ועד לרגע שבו כל התהליכים עוזבים את התור, לא מגיעים לתור תהליכים נוספים.

כמו כן, הניחו כי הזמן הדרוש לביצוע החלפת הקשר (context switch) הוא 0.

התהליך	CPU burst time (msec)
P1	6
P2	15
P3	4
P4	3
P5	8

ט. מה יהיה זמן ההמתנה הממוצע (average waiting time) אם מדיניות התזמון של ה-CPU היא מסוג "ראשון מגיע, ראשון יוצא" (FCFS/FIFO):

1. 7.2 msec
2. 10.4 msec
3. 16.0 msec
4. 36.0 msec

י. מה יהיה זמן ההמתנה הממוצע (average waiting time) אם מדיניות התזמון של ה-CPU היא מסוג "העבודה הקצרה תחילה" (SJF)?

1. 7.2 msec

2. 8.8 msec

3. 16.0 msec

4. 44.0 msec

י"א. הוחלט שהתהליכים יבוצעו לפי עדיפות, ובתוך אותה עדיפות הם יבוצעו לפי שיטת "העבודה הקצרה תחילה" (SJF). לתהליכים P2 ו-P4 נקבעה עדיפות גבוהה (high), ואילו יתר התהליכים הם בעדיפות רגילה (normal). מה יהיה זמן ההמתנה הממוצע (average waiting time) של התהליכים בתור?

1. 14.2 msec

2. 15.1 msec

3. 22.4 msec

4. 28.0 msec

סעיפים י"ב-י"ד מתייחסים לנתונים שלהלן:

הטבלה שלהלן מתארת את תור ההמתנה לקבלת המעבד (ready queue) בנקודת זמן מסוימת t_0 . כל חמשת התהליכים מוכנסים לתור בו זמנית בנקודת זמן t_0 . הם מוכנסים לתור לפי הסדר הבא (מימין לשמאל): A, B, C, D, E. לצד כל תהליך מצוין ה-CPU burst time שלו. כמו כן, בנקודת זמן t_0 , מתזמן המעבד (CPU scheduler) מקצה בפעם הראשונה את המעבד לאחד מחמשת התהליכים שבתור. הניחו כי במהלך כל פרק הזמן שחולף בין t_0 ועד לרגע שבו כל התהליכים עוזבים את התור, לא מגיעים לתור תהליכים נוספים. כמו כן, הניחו כי הזמן הדרוש לביצוע החלפת הקשר (context switch) הוא 0. נתון כי סדר הקצאת המעבד (CPU) הוא בשיטת התזמון Round Robin. יחידת זמן העיבוד המוקצית לתהליך (Time Quantum/ Time Slice) היא של 10 msec. ניהול התור של התהליכים הממתינים לקבלת המעבד (ready queue) הוא מסוג "ראשון מגיע, ראשון יוצא" (FCFS/FIFO).

התהליך	CPU burst time (msec)
A	15
B	25
C	20
D	15
E	35

המעבד משתחרר לקראת ההרצה של התהליך הבא מייד לאחר שסיים את הרצת התהליך הקודם.

י"ב. מה יהיה זמן ההמתנה הממוצע (average waiting time) של התהליכים למעבד?

1. 61.0 msec
2. 65.0 msec
3. 70.0 msec
4. 80.0 msec

י"ג. איזה תהליך, או אילו תהליכים, יימצאו בתור לאחר הפעם העשירית של ריצת מתזמן המעבד?

1. E, D, C, B
2. E, D, C
3. E, B
4. E

י"ד. כמה פעמים יוקצה המעבד לתהליך B במהלך פרק הזמן שבין t_0 לבין $t_0 + 80 \text{ msec}$?

1. 1
2. 2
3. 3
4. 4

סעיף זה אינו קשור לסעיפים הקודמים.

ט"ו. נתון כי תהליך P עובר ממצב running למצב ready. כתוצאה מכך, גרעין מערכת ההפעלה:

1. משחרר את המסגרות שהוקצו לתהליך P ובמקומן מקצה ל-P מסגרות אחרות.
2. משחרר את רשומת בלוק הבקרה (PCB) של התהליך P מטבלת התהליכים.
3. סוגר את הקלט הסטנדרטי ואת הפלט הסטנדרטי של תהליך P.
4. מעדכן את רשומת בלוק הבקרה (PCB) של התהליך P.

פרק שני (18 נקודות)

ענו על שאלה 2, הכוללת שישה סעיפים (א'-ו'). יש לענות על כל הסעיפים. הסעיפים א'-ו' אינם תלויים זה בזה, אלא אם כן מצוין אחרת. בכל הסעיפים נתונות ארבע תשובות, שרק אחת מהן נכונה. בכל סעיף בחרו את התשובה הנכונה, והקיפו בעיגול את הספרה המייצגת אותה בדף התשובות שבנספח א'.

הסעיפים בחלק זה מתייחסים למערכת ההפעלה סולאריס (Solaris).

שאלה 2 (18 נקודות – לכל סעיף 3 נקודות)

א. הפקודה `mv` "מזיזה" קובץ ממקומו המקורי בהיררכיית הקבצים למקום אחר בהיררכיית הקבצים.

מימוש הפקודה `mv /mone/sum/fl /log/yrn`, כרוך ב-

1. הקצאת בלוקי נתונים (data block) חדשים.
2. עדכון טבלת ה-inode.
3. עדכון רשימת הבלוקים הפנויים.
4. עדכון הבעלים (owner) של הקובץ `/log/yrn`.

ב. משתמש מקליד את הפקודה `ps`. איזו שורה יכול הפלט בוודאות?

1. `init`
2. `bash`
3. `login`
4. `ps`

ג. גרעין מערכת ההפעלה (kernel) מפסיק לרוץ. מהי הסיבה האפשרית לתקלה זו?

1. ניצולת המעבד גדולה מדי (מעל 70%).
2. טבלת התהליכים שמנהל הגרעין מלאה לחלוטין.
3. תור התהליכים הממתינים לריצה (ready queue) ריק לחלוטין.
4. יש שגיאת ריצה (bug) בקוד הגרעין של מערכת ההפעלה.

ד. מהו ה־boot loader?

1. סקטור הדיסק שמכיל קוד אשר טוען את תוכנית הגרעין של מערכת ההפעלה (kernel).
2. סקטור הדיסק שבו מנוהלים מבני הנתונים של כל הקבצים הפתוחים (open files).
3. סקטור הדיסק שמשמש לזיכרון המדומה (virtual memory) בדיסק הקשיח.
4. סקטור הדיסק שמכיל קוד אשר טוען את מנהל ההתקן (device driver) של כרטיס הרשת.

ה. לאחר שבוצעה הפקודה **chmod 744 file** – ההרשאות של הקובץ ששמו file יעודכנו להיות:

1. Owner=rwx Group=r-x Other=r-x
2. Owner=rwx Group=r-- Other=r--
3. Owner=rw- Group=r-- Other=r--
4. Owner=r-- Group=r-- Other=r--

ו. משתמש מקליד את הפקודה **~ cd**. למה הדבר יגרום?

1. לאתחול מבנה הנתונים של גרעין מערכת ההפעלה (kernel).
2. לאתחול מבנה הנתונים של ה־bash.
3. לעדכון ספריית העבודה הנוכחית (current working directory) של המשתמש.
4. לעדכון ספריית העבודה הנוכחית (current working directory) של תהליך ה־init.

פרק שלישי (18 נקודות)

ענו על שאלה 3, הכוללת שמונה סעיפים (א'–ח'). יש לענות על שישה מבין הסעיפים (לכל סעיף – 3 נקודות).

שאלה 3 (18 נקודות)

חברת הזנק מפתחת מערכת הפעלה חדשה אשר מיועדת למחשבים ניידים (laptop). המחשב הנייד מכיל מעבד רב־ליבות (Multi-core).

למערכת ההפעלה יש שלושה ממשקי הפעלה שונים: גרפי, טקסטואלי וזיהוי קולי.

לפניכם שמונה סעיפים, שבכל אחד מהם מוצג היגד העוסק במערכת ההפעלה המתוארת לעיל. **בחרו שישה סעיפים, וקבעו לגבי כל אחד מן ההיגדים שבחרתם אם הוא נכון או לא נכון.**

הקיפו בעיגול את התשובה המתאימה בדף התשובות שבנספח א'.

א. תת־המערכת לזיהוי קולי חייבת להיות חלק מהגרעין של מערכת ההפעלה.

ב. הגרעין של מערכת ההפעלה חייב לכלול מתזמן CPU שיודע לנהל מספר ליבות.

ג. הגרעין של מערכת ההפעלה חייב לתמוך בשיטת ניהול זיכרון אשר מותאמת לשיטת ניהול הזיכרון הממומשת בחומרה.

ד. הגרעין של מערכת ההפעלה חייב לכלול מנהל התקן עבור המיקרופון של המחשב שבו מותקנת מערכת ההפעלה.

ה. הגרעין של מערכת ההפעלה חייב לספק שירות של יצירת קובץ חדש.

ו. הגרעין של מערכת ההפעלה חייב לתמוך בשירות של דחיסת וידאו.

ז. מערכת ההפעלה חייבת לתמוך בהרצת תוכניות באצווה (batch).

ח. הגרעין של מערכת ההפעלה חייב לממש מבנה נתונים לשמירת המידע על הבלוקים הפנויים בדיסק הקשיח, כאשר המחשב מכיל דיסק קשיח יחיד.

פרק רביעי (19 נקודות)

ענו על שאלה 4, הכוללת שלושה סעיפים (א'–ג'). יש לענות על כל הסעיפים. הסעיפים א'–ג' אינם תלויים זה בזה. בכל הסעיפים נתונות ארבע תשובות, שרק אחת מהן נכונה. בכל סעיף בחרו את התשובה הנכונה, והקיפו בעיגול את הספרה המייצגת אותה בדף התשובות שבנספח א'.

שאלה 4 (19 נקודות)

(7 נק') א. נתונים שני תהליכים P_0 ו- P_1 .

שני התהליכים משתפים ביניהם את שני המשתנים הבאים:

```
int      turn
boolean flag[2]
```

כמו כן נתון כי הקריאה והכתיבה של ערכי המשתנים הן פעולות המתבצעות באופן אטומי.

לפניכם האלגוריתם שמבצע תהליך P_i :

```
while (true)
{
    flag[i] = TRUE;
    turn = j;
    while (flag[j] && turn == j);
        CRITICAL SECTION
    flag[i] = FALSE;
        REMAINDER SECTION
}
```

האלגוריתם שמבצע תהליך P_i :

1. לא מונע תנאי מרוץ (Race Condition) ולא מונע קיפאון (dead lock).
2. מונע תנאי מרוץ (Race Condition), אך לא מונע קיפאון (dead lock).
3. מונע קיפאון (dead lock), אך לא מונע תנאי מרוץ (Race Condition).
4. מונע תנאי מרוץ (Race Condition), וגם מונע קיפאון (dead lock).

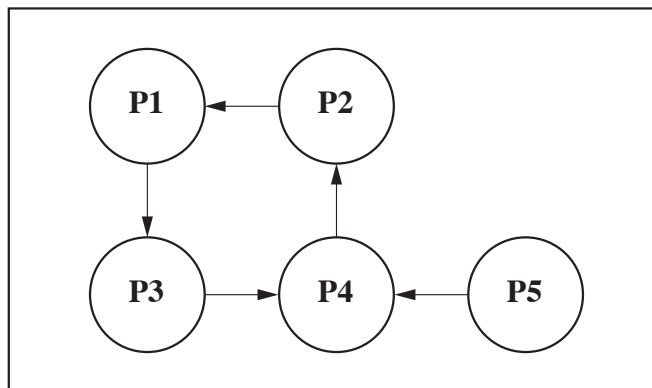
(6 נק')

ב.

במחשב מסוים רצים בו זמנית חמישה תהליכים (קראו מימין לשמאל): P1, P2, P3, P4 ו-P5.

התהליכים משתפים ביניהם משאבים. הדיאגרמה הבאה מתארת את אופן ההמתנה של התהליכים למשאבים.

לדוגמה: תהליך P2 ממתין למשאב שנמצא ברשותו של תהליך P1, תהליך P1 ממתין למשאב שנמצא ברשותו של P3, וכן הלאה.



מה ניתן לקבוע על-פי הדיאגרמה?

1. נוצר מצב קיפאון (dead lock).
2. לא נוצר מצב קיפאון (dead lock).
3. לא ניתן לקבוע חד משמעית אם נוצר מצב קיפאון (dead lock).
4. רק תהליך P1 נמצא במצב קיפאון (dead lock).

ג. (6 נק')

נתונה מערכת הפעלה אשר תומכת בריבוי תהליכים (multi programming).

המחשב שבו מותקנת מערכת ההפעלה כולל, בין היתר, שלושה משאבי חומרה: R1, R2 ו-R3.

הטבלה הבאה מתארת את כמות המשאבים מכל סוג:

משאב	R1	R2	R3
כמות	9	4	7

נתונים ארבעה תהליכים (קראו מימין לשמאל): P1, P2, P3 ו-P4.

התהליכים מתנהגים באופן הבא:

עם תחילת ריצתו, תהליך P_i מבקש ממערכת ההפעלה את כל המשאבים הדרושים לצורך ביצוע משימת החישוב שלו.

רק לאחר קבלת כל המשאבים הדרושים לו, מתחיל P_i בביצוע משימת החישוב שלו.

רק בתום ביצוע משימת החישוב שלו, P_i משחרר את המשאבים שהוקצו לו.

הטבלה הבאה מציגה את המשאבים הנדרשים לכל תהליך לצורך ריצתו:

משאב תהליך	R1	R2	R3
	P1	3	2
P2	5	1	3
P3	1	1	1
P4	4	2	2

מהו מספר התהליכים המקסימלי שיתחילו בביצוע משימת החישוב שלהם באופן **מיידי** (ללא המתנה לקבלת משאבים)?

1. 1

2. 2

3. 3

4. 4

בהצלחה!

לשאלון 714913, אביב תשפ"ג

הדביקו את מדבקת הנבחן שלכם במקום המיועד לכך, והדקו את הדף הזה אל מחברת הבחינה.

שאלה 1 (45 נקודות)					שאלה 2 (18 נקודות)					שאלה 3 (18 נקודות)					שאלה 4 (19 נקודות)				
סעיף א	1	2	3	4	סעיף א	1	2	3	4	סעיף א	1	2	3	4	סעיף א	1	2	3	4
סעיף ב	1	2	3	4	סעיף ב	1	2	3	4	סעיף ב	1	2	3	4	סעיף ב	1	2	3	4
סעיף ג	1	2	3	4	סעיף ג	1	2	3	4	סעיף ג	1	2	3	4	סעיף ג	1	2	3	4
סעיף ד	1	2	3	4	סעיף ד	1	2	3	4	סעיף ד	1	2	3	4	סעיף ד	1	2	3	4
סעיף ה	1	2	3	4	סעיף ה	1	2	3	4	סעיף ה	1	2	3	4	סעיף ה	1	2	3	4
סעיף ו	1	2	3	4	סעיף ו	1	2	3	4	סעיף ו	1	2	3	4	סעיף ו	1	2	3	4
סעיף ז	1	2	3	4						סעיף ז	1	2	3	4	סעיף ז	1	2	3	4
סעיף ח	1	2	3	4						סעיף ח	1	2	3	4	סעיף ח	1	2	3	4
סעיף ט	1	2	3	4															
סעיף י	1	2	3	4															
סעיף י"א	1	2	3	4															
סעיף י"ב	1	2	3	4															
סעיף י"ג	1	2	3	4															
סעיף י"ד	1	2	3	4															
סעיף ט"ו	1	2	3	4															

נספח ב': מילון מונחים
לשאלון 714913, אביב תשפ"ג

המונח	תרגום המונח		
	ערבית	רוסית	אנגלית
אלגוריתם סטטי	خوارزمية ثابتة	Статический алгоритм	static algorithm
אתחול	إبتداء	Загрузка/инициализация	initialization
בלוק	كتلة	Блок	block
גיבוב	مزيج	Хеширование	hashing
החלפת דפים	إستبدال صفحات	Замена страницы	page replacement
היררכיה	هرمية	Иерархия	hierarchy
הקצאה	تخصيص	Назначение	allocation
הרשאת גישה	إذن وصول	Доступ	access permission
השהיה	بقاء	Пауза	pause
זיכרון מטמון	ذاكرة التخزين المؤقت	Кэш-память	cache memory
חיפוש בינארי	بحث ثنائي	Двоичный поиск	binary search
חיפוש סדרתי	بحث تسلسلي	Серийный поиск	sequential search
חריג	شاذ	Аномальный	exception
טעינה	تحميل	Загрузка	loading
יחס החטאה	نسبة الإخطاء	коэффициент кэш-промахов	miss ratio
כשל דף	فشل الصفحة	Отказ в доступе к странице	page fault
מחיצה	فاصل	раздел	partition
מטריצה	مصفوفة	Матрица	matrix
מיפוי	رسم الخرائط	Картирование	mapping
מניעה הדדית	إقصاء متبادل	Взаимное исключение	mutual exclusion
מערכת מבוזרת	نظام موزع	Децентрализованная система	distributed system
סנכרון	تزامن	Синхронизация	synchronization
פסיקה	قطع	Прерывание	interrupt
קובץ	ملف	Файл	file
קטע קוד קריטי	مقطع الكود الحرج	Критический участок кода	critical code section
קיפאון	ركود	Застой	deadlock
שיטת זיכרון מדומה	محاكاة نظام الذاكرة	Система виртуальной памяти	virtual memory method
תהליך אצווה	معالجة بدفعات	Пакетная обработка	batch process
תזמון	توقيت	Синхронизация	timing