

אוסף תרגילי בית תיאורטיים

הוראות כלליות

- יש להגיש את פתרונות התרגילים **מודפסים**. העמוד הראשון יהיה "עמוד שער" כמצורף לתרגיל ויש למלא אותו בכתב ברור.
 - עבור שאלות אמריקאיות: עבור כל אפשרות יש להוכיח את קיומה או להפריך אותה.
 - ההגשה תיעשה ביחידים בלבד. אין להגיש בזוגות או בקבוצות גדולות יותר.
1. הנך מהנדס/ת תוכנה שתפקידו/ה לבחור מערכת הפעלה לפרוייקט מסוים. אילו שירותים היית מצפה שמערכת ההפעלה תספק אם הפרוייקט הוא:
 - א. תוכנית מעבד תמלילים.
 - ב. תוכנית גיליון נתונים.
 - ג. מערכת נתונים שתנהל את שירותי מערכת הבריאות למדינת ישראל (מס' תושבים כ-6 מיליון).
 - ד. מכונה למכירת שתייה קלה.
 - ה. משחק מחשב המשתמש בגרפיקה תלת-ממדית.
 2. אילו סיבות משפיעות על יצרני מערכות הפעלה בנוגע לחלקים שעליהם לכלול וחלקים שלא לכלול במערכת ההפעלה שלהם?
 3. (אוני' סידני) חלק מהמאפיינים של מערכת הפעלה טובה הם: יעילות, אמינות, בטיחות, נוחות ו"חברותית" (בעלת יכולת לתקשר עם מערכות אחרות). האם יש להוסיף או להחסיר מאפיינים מהרשימה הנ"ל? לעתים, חלק מהמאפיינים מתנגשים זה בזה. תנו דוגמאות למקרים כאלו.
 4. (אוני' סידני) מדוע מערכות הפעלה נוטות להיות עם הרבה "באגים"?
 5. מה ההבדל בין פסיקה למלכודת (trap)?
 6. הסבירו (ע"י שרטוט ו/או מלים) מה קורה כאשר אחת מהפסיקות הבאות קורות כאשר המעבד מתבצע במצב משתמש (user mode). מתי פסיקות אלו יכולות לקרות? כיצד מערכת ההפעלה מטפלת בהן?
 - א. קריאות מערכת.
 - ב. פסיקות חיצוניות.
 - ג. מלכודות (traps).
 7. (אוני' סידני) כיצד פעולות ק/פ שהן לא-מקוונות (offline) תורמות לביצועים? מה מרוויחים ממעקפי קלט ומעקפי פלט?
 8. הסבירו את המושגים הבאים (בהקשר של מערכות הפעלה). השוו והבדילו ביניהם (תוך שימוש בטבלה):
 - א. אצווה (batch).
 - ב. מקוון (online).
 - ג. הידברותי (interactive).
 - ד. שיתוף זמנים (time-sharing).
 - ה. multiprogramming.
 - ו. זמן-אמת (real-time).
 - ז. מבוזר (distributed).

9. (אוני' סידיני) מה המוטיבציה מאחורי תכנון מערכת הפעלה תוך שימוש בשכבות (layers)? לאחרונה, יש נטייה לצמצם את מספר השכבות תוך הגדלת הפונקציונליות של כל שכבה. מדוע?
10. אם מרחב הזיכרון של המעבד הוא לינארי, כלומר אין מקטוע (segmentation), העלאת מספר תוכניות לזיכרון בבת-אחת גורם לבעיות שונות. הציעו פתרון שיאפשר למערכת ההפעלה להעלות תוכנית למקום שרירותי בזיכרון ולהריץ אותה שם.
11. מערכות הפעלה שתומכות בריבוי תהליכים (multitasking) מומשו בעבר גם עבור מיקרו-מעבדים שלא היה להם הגנת זיכרון. אילו דרישות צריך לכפות על תוכניות שירוצו במערכת הפעלה כזו? האם ניתן לתכנן מערכת הפעלה כזו כך שתוכניות יהיו מוגנות בצורה טובה אחת מהשנייה?
12. הסבירו באיזה מהמאורעות הבאים כדאי לטפל ישירות בטכניקות של פסיקה חיצונית (אסינכרונית)?
- א. איתות מעמדת המפעיל.
 - ב. ניסיון לביצוע פקודה לא חוקית.
 - ג. גישה לכתובת פיסית שלא קיימת במחשב.
 - ד. תקלת חומרה במחשב.
 - ה. נפילת מתח קרובה.
13. הסבירו באיזה מהמאורעות הבאים כדאי לטפל ישירות בטכניקה של מלכודת (trap)?
- א. תהליך נמצא במצב של לולאה אינסופית.
 - ב. תהליך מבקש יותר זיכרון מאשר המערכת יכולה לספק לו.
 - ג. שגיאת ק/פ.
 - ד. גישה למשתנה שאין לו ערך תחילי (not initialized).
 - ה. שמירת מספר גדול יותר מאשר המערכת יכולה לאחסן במילה (overflow).
14. ישנן 2 שיטות אפשריות לסימון מצבו הנוכחי של תהליך:
- א. ערך מתאים בשדה המצב בכניסת התהליך (PCB), לדוגמה: מצב מוכן (Ready).
 - ב. שילוב/שרשור בכניסת התהליך ברשימת התהליכים הנמצאים באותו מצב מסוים, לדוגמה תור המוכנים (ready queue).
- מהם היתרונות והחסרונות (בצורת טבלה) של כל אחת משתי שיטות אלו? האם יש טעם בשימוש בשתייהן במקביל? הדגימו את טיעוניכם.
15. המצבים הקלאסיים של תהליך הם:
- מוכן - ready, ממתין - waiting, רץ - running.
- אולם יש גם מספר סיבות לצורך בהשיית תהליך לזמן מסוים, לכן דרושות שתי פעולות נוספות: השהה - suspended, והמשך - resume. לתמיכה בהשהיה מוצע להוסיף 4 מצבי תהליך נוספים:
- מוכן-מושהה ready-suspended
ממתין-מושהה waiting-suspended
רץ-מושהה running-suspended
לא מושהה resumed
- איזה ממצבים אלו באמת דרוש במערכת? מנו (בקצרה) מספר סיבות שונות אפשריות לכניסה לכל אחד מהמצבים החדשים הדרושים. שרטטו את גרף המעברים (state transition) המורחב בין המצבים של התהליך (כולל הסבר למעבר על כל קשת).

16. אחת מהבעיות בניהול מדרג של תהליכים הוא סיום תהליך. לגבי תהליך הורה (parent) שיצר תהליכי ילד (child), ישנן 3 אפשרויות:
א. אסור להורה לסיים או להיהרג לפני סיום או הרג מפורש של ילדיו.
ב. סיום/הרג הורה פירושו גם הרג אוטומטי של כל ילדיו (cascading termination).
ג. ההורה מסיים/נהרג, אך ילדיו ממשיכים לרוץ.
- השוו בין 3 האפשרויות תוך שימוש בטבלה ששורותיה: תלות בקיום ההורה, דיווח סיום ילד להורה, טיפול בשטחי code/data משותפים, תמיכת מערכת הפעלה, תמיכת משתמש/מתכנת, גמישות השיטה.
איזו שיטה מיושמת ב-Unix?
17. לגבי חסימת פסיקות (Interrupt Disable) אפשר לטעון ש:
א) חסימת פסיקות מונעת מהמעבד לעבור לטיפול בתהליך אחר וגורמת לקפאון (Deadlock).
ב) זוהי שיטה המבטיחה מניעה הדדית בחד-מעבד (Uniprocessor) וברב-מעבד (Multiprocessor).
ג) פסיקות לא ילכו לאיבוד שכן הסימון שהתרחשו פסיקות יישמר.
ד) היתרון של גישה זו הינו פשטות ושימוש באמצעי חומרה קיימים.
ה) כל התשובות המפורטות צריכות להיות מסומנות.
ו) אף תשובה מפורטת אינה צריכה להיות מסומנת.
18. האם תתכן מקביליות אמיתית (True Concurrency) (כלומר ביצוע יותר מפעולה אחת בו זמנית) במחשב חד-מעבד (Uniprocessor)?
א) כן.
ב) לא. זוהי אשליה הנוצרת בגלל פער המהירות בין פעולות ק/פ לבין מהירות מעבד.
ג) לא. המקביליות המדומה מתאפשרת בגלל היכולת לשחזר בדיוק מחדש תהליך שהופסק זמנית.
ד) לא. כי המעבד יכול לשרת רק תהליך אחד בכל "רגע".
ה) כל התשובות המפורטות צריכות להיות מסומנות.
ו) אף תשובה מפורטת אינה צריכה להיות מסומנת.
19. לגבי פסיקה חיצונית (External Interrupt) אפשר לטעון ש:
א) ניתן לחזות מראש את הזמן בו תתרחש הפסיקה החיצונית הבאה.
ב) פסיקה חיצונית אינה יכולה להתרחש באמצע טיפול בפסיקה חיצונית קודמת.
ג) הודעה של התקן ק/פ כלשהו על סיום פעולת ק/פ אינה פסיקה חיצונית.
ד) פקודת מכונה להפסיק (Halt) מחשב גורמת לפסיקה חיצונית.
ה) כל התשובות המפורטות צריכות להיות מסומנות.
ו) אף תשובה מפורטת אינה צריכה להיות מסומנת.
20. לגבי שימוש בתהליכים ברמת הגרעין (Kernel supported threads) לעומת תהליכים אפשר לטעון ש:
א) יצירת תהליכון, בתוך תהליך הורה (משימה) קיים, מחייבת אותו מאמץ כמו יצירת תהליך חדש.
ב) למיתוג הקשר בין שני תהליכים השייכים לאותו תהליך הורה תקורה נמוכה יותר מאשר למיתוג הקשר בין שני תהליכים שונים.
ג) הקשר (Context) תהליכון זהה להקשר תהליך.
ד) מבחינת תקורת זמן אין הבדל בין תהליכים לתהליכים, החסכון הוא בתקורת הזיכרון.
ה) כל התשובות המפורטות צריכות להיות מסומנות.
ו) אף תשובה מפורטת אינה צריכה להיות מסומנת.

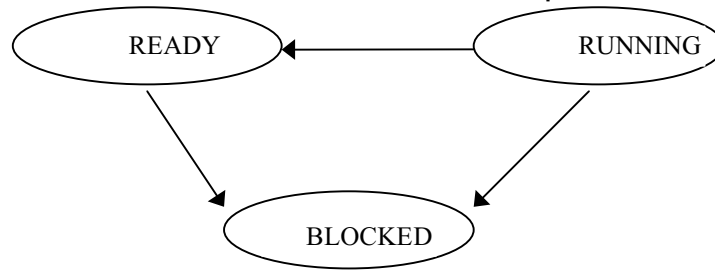
21. ניתן לממש סמפור בעזרת מונה ותור. לגבי ניהול התור במשטר "אחרון נכנס ראשון יוצא" (LIFO) אפשר לטעון ש:
- א) משטר כזה אינו הוגן.
 - ב) מובטחת המתנה חסומה (Bounded Wait).
 - ג) אפשר להשתמש בסמפור כזה לפתרון בעיית הקטע הקריטי.
 - ד) תהליכים יטופלו לפי סדר ההגעה.
 - ה) כל התשובות המפורטות צריכות להיות מסומנות.
 - ו) אף תשובה מפורטת אינה צריכה להיות מסומנת.

22. תהליך יכול להיות באחד משלושת המצבים הבאים :

RUNNING ←
READY ←
BLOCKED ←

הגדר כל אחד ממצבים אלה.

הסבר את המעברים המסומנים (ורק אותם) בדיאגרמה הבאה. אם אחד או יותר מהמעברים אינו אפשרי, ציין מדוע הוא אינו אפשרי.



23. בתכנית הבאה מוצג פתרון לבעיית היצרן-צרכן:

```
#define N 100
```

```
typedef int semaphore;
semaphore mutex = 1;
semaphore empty = N;
semaphore full = 0;
```

```
void producer(void) {
{
    int item;
    while (TRUE) {
        produce_item(&item);
        down(&empty);
        down(&mutex);
        enter_item(item);
        up(&mutex);
        up(&full);
    }
}
```

```
void consumer(void) {  
    {  
        int item;  
        while (TRUE) {  
            down(&full);  
            down(&mutex);  
            remove_item(&item);  
            up(&mutex);  
            up(&empty);  
            consume_item(item);  
        }  
    }  
}
```

נניח שבתכנית הבאה מחליפים את סדר הפעולות על הסמפורים בתכנית היצרן בלבד.
כלומר תוכנית היצרן היא:

```
void producer(void) {  
    {  
        int item;  
        while (TRUE) {  
            produce_item(&item);  
            down(&mutex);  
            down(&empty);  
            enter_item(item);  
            up(&full);  
            up(&mutex);  
        }  
    }  
}
```

א. האם התכנית המתקבלת עדיין פותרת את הבעיה? בחרו את התשובה המתאימה ביותר.
א) כן.
ב) כן, אבל רק במערכת עם מעבד יחיד (uniprocessor).
ג) לא.
ד) לא. יש סכנת קיפאון.

ב. הפתרון לבעיית היצרן-צרכן המוצג למעלה (הפתרון המקורי, לפני השנוי) ניתן לשימוש גם למקרה ש: (בחרו את התשובה המתאימה ביותר).
א) יש צרכנים רבים אך לא יותר מיצרן אחד.
ב) יש יצרנים רבים אך לא יותר מצרכן אחד.
ג) יש יצרנים רבים וגם צרכנים רבים אך במספר שווה.
ד) יש יצרנים רבים וגם צרכנים רבים (במספר כלשהו).
ה) כל התשובות המפורטות צריכות להיות מסומנות.
ו) אף תשובה מפורטת אינה צריכה להיות מסומנת.

24. תוכנית מתבקש להפעיל שתי תוכניות בזו אחר זו: PROG1 ו-PROG2. לשם כך הוא כתב את התוכנית הבאה:

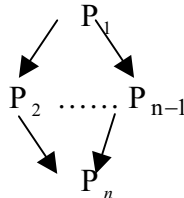
(הניחו ש- exec1 היא פונקציה ממשפחת ה-exec ושבתוכנית אין "שגיאת C")

```
main ()  
{  
    /* do what ever is needed before doing PROG1 */  
    .....  
    /* now execute PROG1 */  
}
```

```
excel("PROG1", "PROG2", NULL);  
/* now execute PROG2 */  
excel("PROG2", "NEW-NAME-FOR-PROG2", NULL);  
/* do what ever is needed to finish the program */  
.....  
}
```

האם התכנית הנ"ל משיגה את המטרה? הסבירו כיצד! מה בדיוק קורה?

25. במערכת בת n תהליכים, נתון גרף הקדימויות הבא:



כלומר P_2, \dots, P_{n-1} יכולים להתבצע (במקביל) רק לאחר סיום P_1 , ו- P_n מתבצע רק לאחר שכל שאר התהליכים סיימו לעבוד. כתוב מימוש של גרף הקדימויות הנ"ל ע"י שימוש בסמפורים (יש לציין רק את הפעולות על הסמפורים בכל תהליך). חפשו פתרון המשתמש במספר מינימלי של סמפורים.

26. בין הטיעונים השונים לגבי סמפורים ניתן (אולי) לטעון:

- שתי פעולות P המופעלות על שני סמפורים שונים חייבים להיות "מנועים הדדית".
 - הערך ההתחלתי של סמפור לא יכול להיות אפס.
 - הערך ההתחלתי של סמפור לא יכול להיות שלילי.
 - אחרי אתחול סמפור אין לשנות את ערכו אלא ע"י הפעולות P ו- V .
 - מכניזם היישום של סמפור חייב להשתמש בתור FIFO.
 - אין טעם לספק פעולות תיחול (initialize) וקריאת ערך של סמפור כפעולות בסיסיות של סמפור.
- אלו מהטיעונים נכונים? הסבירו.

27. לעיתים נמצא במחשב שימוש הן בהוראת `test & set` והן בסמפורים. כל אחד ממלא תפקיד שונה ואין הם מתחרים האחד בשני. הסבירו מדוע?

28. מהו ההבדל בין סמפור בינרי לסמפור כללי? הדגימו שימוש לכל אחד מסוגי סמפורים אלו. האם סוג סמפור אחד יכול לשמש חלופה לסוג השני? הסבירו.

29. מוצע הפתרון הבא לפתרון הקטע הקריטי (שני אנשים: JIM ו-LARRY מנסים לגשת לאותו קטע קריטי):

```
FLAG_LARRY := true  
While TURN is not equal to "Larry" do  
  Begin  
    While FLAG_JIM is true, meditate  
    TURN := "Larry"  
  End  
Access other shared data  
FLAG_LARRY := false
```

האם ההצעה הנ"ל פותרת את בעיית הקטע קריטי? באילו הנחות? (הראו מדוע ההנחות שהנחתם, אם בכלל, חיוניות). לגבי כל אחת משלוש הדרישות לקיום פתרון, או הוכיחו את קיומה או לחליפין הראו דוגמא שמפירה אותה.

30. מוצע הפתרון הבא לפתרון הקטע הקריטי של Jim (שני אנשים: JIM ו-LARRY מנסים לגשת לאותו קטע קריטי):

```
FLAG_LARRY := true
While FLAG_JIM is true do
  if TURN = "Jim" then begin
    FLAG_LARRY := false
    while TURN = "Jim", meditate
    FLAG_LARRY := true
  End
  Access other shared data
TURN := "Jim"
FLAG_LARRY := false
```

האם ההצעה הנ"ל פותרת את בעיית הקטע קריטי? באילו הנחות? (הראו מדוע ההנחות שהנחתם, אם בכלל, חיוניות).
לגבי כל אחת משלוש הדרישות לקיום פתרון, או הוכיחו את קיומה או לחליפין הראו דוגמא שמפירה אותה.

31. מוצע הפתרון הבא לפתרון הקטע הקריטי של Larry (שני אנשים: JIM ו-LARRY מנסים לגשת לאותו קטע קריטי):

```
var FLAG_LARRY: (idle, want-in-cs, in-cs) (* idle is default *)
repeat
  FLAG_LARRY := want-in-cs
  While (TURN <> "Larry") and (FLAG_JIM <> idle), meditate
  FLAG_LARRY := in-cs
until (FLAG_JIM <> in-cs)
TURN := "Larry"
  Access other shared data
TURN := "Jim"
FLAG_LARRY := idle
```

האם ההצעה הנ"ל פותרת את בעיית הקטע קריטי? באילו הנחות? (הראו מדוע ההנחות שהנחתם, אם בכלל, חיוניות).
לגבי כל אחת משלוש הדרישות לקיום פתרון, או הוכיחו את קיומה או לחליפין הראו דוגמא שמפירה אותה.

32. מוצע למנוע קיפאון בבעיית הפילוסופים הסועדים תוך שימוש בטכניקה של מניעת קיפאון (deadlock prevention). תנו 4 פתרונות המונעים כל אחד בתורו קיום של אחת מארבע הדרישות המוקדמות להיווצרות קיפאון. הסבירו.

33. מה השווה והשונה (בצורת טבלה) בין התופעות של קיפאון ושל הרעבה? תנו דוגמאות מסוימות (באותו תחום) לבעיות קיפאון והרעבה המבליטות את השוני ביניהן.

34. במערכת ויסות מעבד השתמשנו במושגים עבודה, תהליך והדגם (instance). מה השווה והשונה בין שלושת המושגים האלו? באיזה ממרכיבי המערכת באים לידי ביטוי ההבדלים ביניהם? הסבירו.

35. באלגוריתם ויסות מעגלי (round robin) מצרפים תהליכים חדשים לסוף התור. הצירוף עצמו יכול להיעשות עם גמר פלח הזמן (כולו או חלקו) של התהליך האחרון שרץ. לפיכך ישנן 2 אפשרויות:

- א. התהליכים החדשים יכולים להצטרף לתור (לפי סדר הגעתם) לפני התהליך שזה עתה סיים (ולכן הוא יהיה האחרון בתור).
 ב. אפשר לצרף את החדשים אחריו בתור.
 השוו (בצורת טבלה) בין 2 גישות אלו לפי המאפיינים והשיקולים שעשינו בהם שימוש להשוואת אלגוריתמים בויסות מעבד.

36. באלגוריתם ויסות מעבד לפי Round Robin נתונה רשימת התהליכים הבאה:

| תהליך | זמן הגעה | פרץ עיבוד |
|-------|----------|-----------|
| A | 0 | 5 |
| B | 2 | 3 |
| C | 3 | 2 |
| D | 6 | 3 |

השוו (ע"י תרשים גאנט וממוצע זמן סבב) בין שתי השיטות של צירוף לתור round robin כמוסבר בשאלה הקודמת.

37. באלגוריתם ויסות מעבד SJF משתמשים בנוסחה שלמדנו להערכת פרץ עבור $\alpha = 1/2$. בשלב מסוים, לתהליך מסוים, משתווה משך פרץ העיבוד להערכת פרץ העיבוד של 15 (יחידות זמן). אולם, אח"כ, מתברר שמשכי פרצי העיבוד של (מופעי) התהליך הם (משמאל לימין): 25, 40, 60. מהי ההערכה של SJF לפרץ העיבוד הבא של התהליך הזה?

38. בויסות מעבד לא רצוני, משתמשים באלגוריתם ויסות לפי עדיפות. כל תהליך מגיע עם עדיפות התחלתית. עם הזמן משנים את עדיפותו בצורה הבאה:
 א. כל T1 יחידות זמן מוסיפים A לעדיפות של כל התהליכים הממתינים (לא רצו בזמן T1).
 ב. כל תהליך שמבקש ק/פ בזמן ריצה מורידים מעדיפותו B.
 ג. כל תהליך ששה במערכת יותר מ-T2 יחידות זמן מוסיפים לעדיפותו C.
 ד. כל תהליך חדש מקבל תוספת D.
 מה המשמעות של כל אחד מהשינויים א'-ד'? מה תהייה ההשפעה של הקטנת/הגדלת A-D (השאיפו אותו ל-0 ולא ינוסף).

39. נתונה מערכת זיכרון וירטואלי עם דפדפוף. הזיכרון הפיסי מכיל 8 מסגרות, כל אחת בגודל 4 בתים. נתונה תוכנית בעלת 12 דפים לוגיים. אילו מהמיפויים הבאים, בין מען לוגי למען פיסי (יכולים להיות) חוקיים (התייחסו לכל מיפוי בנפרד) בהקשר הנתון?

| | מען פיסי | מען לוגי |
|---|----------|----------|
| א | 31 | 27 |
| ב | 17 | 57 |
| ג | 1 | 29 |
| ד | 37 | 41 |
| ה | 12 | 14 |

40. במערכת זיכרון וירטואלי עם דפדוף נתונה מחרוזת ההתייחסות לדפים הבאים (משמאל לימין): 1, 2, 3, 4, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 9, 5, 4, 5, 4, 2, FIFO, מהו מספר תקלות הדף ומספר החלפות הדף שיקרו לאלגוריתמי החלפה של LRU ו-OPT בהנחה של 4 מסגרות בזיכרון הפיסי (שריקות מלכתחילה).

41. במערכת דפדוף ראינו את השימוש האפשרי בסיבית התייחסות (reference bit) ובסיבית לכלוך (dirty bit). יש אפשרות לשים סיביות אלו בכל כניסת דף בכל טבלת דף (לכל תהליך), או לשים אותם בכל כניסת מסגרת בטבלת המסגרות של המערכת. השוו (בצורת

טבלה) בין 2 אפשרויות אלו. מה היתרונות והחסרונות של כל אחת משיטות אלו. קחו בחשבון גם את צורת היישום, המחיר שלו, שיתוף דפים, הגנת דפים וכו'.

42. לגבי הטיעונים לגבי signals ב-Unix אפשר (אולי) למנות:
א. קריאות המערכת (system calls) של sockets מיושמים ע"י שימוש ב-signals.
ב. מלכודת (trap) ב-Unix מיושמת ע"י שימוש ב-signals.
ג. יש הבדל בין SIGSTOP לבין SIGKIL.
ד. אין בכלל קשר בין signalling context של תהליך הילד ושל תהליך ההורה שלו.

43.
א. הגדר "race condition" ותן דוגמה למצב כזה.
ב. נכון או לא נכון: race condition הוא סוג של אי-דטרמיניסטיות בתוכנית.

44.
א. הגדר עבודה (job), משימה (task) ותהליכון (thread).
ב. מה ההבדל בין תוכנית לתהליך?
ג. מה ההבדלים בין תהליך, LWP ותוכנית? (השתמשו בטבלה).

45. הגדר: zombie (בהקשר לתהליכים ב-Unix).

46. (מבחן במערכות הפעלה / דר' סיון טולדו, פרופ' חזי ישורון, אוני' ת"א)
קטע הקוד הבא מיישם מבנה נתונים של תור FIFO תוך שימוש ברשימה מקושרת דו-צדדית. התור מכיל מספרים שלמים אי-שליליים והוא מיועד לשימוש בתוכניות עם מספר תהליכונים ולכן הוא משתמש במנעולים (mutexes) (מנעול שקול לסמפור בינרי, כאשר הפעולה lock שקולה ל-down והפעולה unlock שקולה ל-up).

```
typedef struct {
    queue_element* next;
    queue_element* prev;
    int data;
    mutex m;
} queue_element;

queue_element* head = null;
queue_element* tail = null;

mutex head_m;
mutex tail_m;

fifo_put(int d)
{
    lock(head_m);
    queue_element* new = malloc(sizeof(queue_element));
    new -> data = d;
    new -> prev = null;
    new -> next = head;
    if (new -> next != null) {
        lock(new -> next);
        new -> next -> prev = new;
        unlock(new -> next);
    } else {
```

```
        lock(tail_m);
        tail = new;
        unlock(tail_m);
    }
    unlock(head_m);
}

fifo_get()
{
    int d; /* return value */
    lock(tail_m);
    if (tail == null) {
        d = -1; /* signifies that the queue is empty */
    } else {
        queue_element* temp = tail;
        d = temp -> data;
        if (tail -> prev != null) {
            lock(tail -> prev -> m);
            tail -> prev -> next = null;
            tail = tail -> prev;
            unlock(tail -> m);
        } else {
            lock(head_m);
            head = null;
            tail = null;
            unlock(head_m);
        }
        unlock(temp -> m);
        free(temp);
    }
    unlock(tail_m);
    return d;
}
```

- א. מדוע איננו נועלים את איבר הרשימה החדש `new` ברוטינה `fifo_put`?
- ב. שימוש בשגרות אלה עלול להביא לקיפאון. הראה תסריט שבו שני תהליכונים נכנסים לקיפאון.
- ג. הצע פתרון פשוט שימנע קיפאון.

47. (מבחן במערכות הפעלה / דר' סיון טולדו, פרופ' חזי ישורון, אוני' ת"א)

במערכת עם דפדוף לפי דרישה (demand paging) נמדדו:

א. CPU Utilization = 25%

ב. Paging disk Utilization = 90%

ג. Other I/O Utilization = 6%

איזה מהשינויים הבאים ישפר את ניצולת המעבד:

(א) מעבד (CPU) מהיר יותר.

(ב) רשת תקשורת מהירה יותר.

(ג) גודל דף (page size) גדול יותר.

48. (הטכניון)

נתון בניין רב-קומות ובו N משפחות. לבניין כולו מסופקים M יחידות חשמל. כל משפחה צריכה K יחידות חשמל כדי להכין את ארוחת הערב. כל משפחה יכולה לקבל או לשחרר יחידת חשמל אחת בלבד בכל רגע נתון.

א. מה הבעיה שיכולה להתעורר בהקצאת יחידות החשמל למשפחות?

ב. רשום אלגוריתם עבור משפחה תוך שימוש במספר סמפורים מינימלי.

הערה: עליך להבטיח שכל משפחה תוכל להכין ארוחת ערב תוך זמן סופי, וכן

שבו-זמנית הבניין לא יצרך יותר מ- M יחידות חשמל. לשם ביצוע המשימה עומדים

לרשותך סמפורים הוגנים.

ג. עבור אילו ערכים של M , כתלות ב- N וב- K , לא מתעוררת הבעיה שציינת בסעיף א'.

49. (הטכניון)

במספרה 2 חדרים: חדר הספר וחדר המתנה בו N כיסאות. הדלת המפרידה ביניהם ודלת הכניסה הראשית מאפשרות מעבר רק לאדם אחד בזמן נתון. כמו כן, בכל רגע נתון, מקסימום אחת מ-2 דלתות אלו פתוחות (ולא שתיהן). כאשר הספר מסיים טיפולו בלקוח הוא מתיר ללקוח הבא להיכנס לחדרו. אם חדר המתנה ריק, הספר מתיישב ונרדם. אם מגיע לקוח ורואה את הספר ישן, הוא מעיר אותו (בעדינות). אחרת, הוא ממתין לתורו בחדר המתנה.

הדמה את המספרה באמצעות סמפורים.

50. (הטכניון)

תוכנית המורכבת מ- $n + 1$ דפים, אחד לקוד ו- n לנתונים, רצה במערכת זיכרון וירטואלי שעובדת בדפדוף לפי דרישה (Demand Paging). המחשב מקצה לתוכנית בזיכרון הפיסי X דפים.

התוכנית עצמה עוברת על הדפים הנתונים אחד אחרי השני n פעמים באופן הבא:

בהתחלה המחשב עובר על כל הדפים D_1, D_2, \dots, D_n ; לאחר מכן הוא שוב עובר על

$n-1$ דפים מהדף השני, כלומר D_2, D_3, \dots, D_n ; בשלב ה- i המחשב עובר על דפים

D_i, \dots, D_n .

כמה תקלות דף יהיו בכל אחת מהשיטות הבאות:

| השיטה | הנוסחה |
|-------------------|--------|
| LRU | |
| FIFO | |
| (OFFLINE) OPTIMAL | |

51. (הטכניון)

נתונים 8 תהליכים שעובדים במחזוריות אינסופית על-פי הסכמה הבאה:

P_0 מתחיל ו- P_1, P_2, P_3 יכולים להתחיל רק אחרי ש- P_0 סיים.

P_4 מתחיל רק אחרי ש- P_1 ו- P_2 סיימו.

P_5 מתחיל רק אחרי ש- P_2 ו- P_3 סיימו.

P_6 מתחיל רק אחרי ש- P_1 ו- P_3 סיימו.

P_7 מתחיל רק אחרי ש- P_4, P_5, P_6 סיימו.

המחזור מתחיל מחדש ו- P_0 חוזר שוב להתבצע רק אחרי ש- P_4 מתחיל רק אחרי ש- P_7 סיים.

תוך שימוש בסמפורים כתוב קוד ל- P_0, P_1, P_5, P_7 . מה ערכם ההתחלתי של כל אחד

המספורים בקוד?

52. להלן זמני הגעה וביצוע של קבוצת תהליכים שמגיעים לתזמון. זמני הביצוע ידועים מראש למתזמן:

| זמן הגעה | זמן ביצוע | |
|----------|-----------|----|
| 0 | 12 | J1 |
| 2 | 8 | J2 |
| 3 | 4 | J3 |
| 7 | 9 | J4 |
| | | . |
| | | . |
| | | . |

תהליכים מגיעים כל 35 שניות (כלומר, J5 מגיע בזמן 35 ולוקח 12 שניות, J6 מגיע בזמן 37 ולוקח 8 שניות, וכן הלאה).

1. עבור FCFS: קבע את זמן התגובה הממוצע (average response time) והמספר המקסימלי של תהליכים במערכת בכל זמן.
2. כנ"ל עבור SJF ללא לקיחה בכוח.
3. כנ"ל עבור SJF עם לקיחה בכוח.
4. כנ"ל עבור האלגוריתם האופטימלי (offline).

53. (University of Maryland)

1. מהם 4 התנאים ההכרחיים לקיפאון?
2. מדוע תנאים אלו הכרחיים אבל לא מספיקים?

54. (University of Maryland)

- א. בהנחה שבמחשב קיימת פעולה אטומית swap, אבל לא קיימת הפעולה test-and-set. הראו כיצד ניתן לדמות את הפעולה test-and-set ע"י שימוש ב-swap.
- ב. במערכת רב-מעבד, מדוע אין זה מספיק להתעלם מפסיקות של מעבד אחד כפתרון אפשרי לבעיית הקטע הקריטי?

55. (University of Maryland)

- כתוב אלגוריתם שפותר את הבעיה הבאה:
יש תהליכים קוראים ויש תהליכים כותבים. עד 5 תהליכים קוראים יכולים לקרוא בו-זמנית, אבל רק תהליך כותב אחד יכול לכתוב בזמן נתון. כתיבה וקריאה באותו זמן אסורה.
על האלגוריתם להשתמש בסמפורים ולקיים את התכונות הבאות:
- א. אין המתנה-מעסיקה (busy-waiting).
 - ב. אין הרעבה.
 - ג. לא ניתן להשתמש במזהים עבור התהליכים ולא ניתן להשתמש בסמפור נפרד עבור כל תהליך.

56. (University of Maryland)

- כתוב אלגוריתם שפותר את הבעיה הבאה:
יש תהליכים קוראים ויש תהליכים כותבים. תהליכים קוראים יכולים לקרוא בו-זמנית, ותהליכים כותבים יכולים לכתוב ב-זמנית. כתיבה וקריאה באותו זמן אסורה.
על האלגוריתם להשתמש בסמפורים ולקיים את התכונות הבאות:
- א. אין המתנה-מעסיקה (busy-waiting).
 - ב. אין הרעבה.
 - ג. לא ניתן להשתמש במזהים עבור התהליכים ולא ניתן להשתמש בסמפור נפרד עבור כל תהליך.

57. (University of Maryland)

יש טענה הגורסת שלכל אסטרטגית תזמון קיימת אסטרטגיה נגדית שמנצלת את האסטרטגיה המקורית לטובתה ולרעת המשתמשים האחרים. עבור כל אחת מהאסטרטגיות הבאות מצאו אסטרטגיה נגדית כזו. עבור כל המקרים הניחו שאנו משתמשים ברב-תורי משובי (multi-level feedback queue) שמשתמש בתזמון ויסות מעגלי בכל רמת עדיפות: א. כל תהליך שמשתמש בכל פלח הזמן המוקצה יורד לרמה אחת נמוכה יותר בתור. ב. כל תהליך שמבצע פעולת ק/פ במהלך פלח התזמון שלו מועלה לרמה אחת גבוהה יותר בתור.

דוגמה:

דוגמה נגדית עבור האסטרטגיה הבאה:

כל זמן שנותר בפלח הזמן והתהליך לא השתמש בו מוסף לפלח הזמן הבא שלו. פתרון: התחל את התהליך לפני שהוא צריך לרוץ וגרום לכך שהוא יפסיק את הריצה שלו באופן חוזר על-מנת לקבל עוד זמן בריצה הבאה. כאשר התהליך מוכן לריצה, הוא יוכל להחזיק במעבד להרבה זמן.

58. הגדר את המושג קטע קריטי (3 תנאים).

59. (University of Maryland)

השתמש ב-test-and-set בשביל לכתוב אלגוריתם שפותר את בעיית הפילוסופים הסועדים. על הפתרון לעבוד עבור כל מספר של סועדים וחייב להבטיח שאף פילוסוף לא ימות מרעב.

60. (University of Maryland)

סמפור בינרי יכול לקבל את הערכים 0 או 1. סמפור מונה יכול לקבל ערכים בין 0 ל-n. ע"י שימוש בסמפורים בינריים, הראה כיצד ניתן לממש סמפור מונה. יש לציין את הסמפורים הבינריים והמשתנים בהם השתמשתם, וכן את ערכי ההתחלה שלהם.

61. (University of Maryland)

סמפור בינרי יכול לקבל את הערכים 0 או 1. סמפור מונה יכול לקבל ערכים בין 0 ל-n. ממש סמפור בינרי ע"י סמפורים מונים. יש לציין את הסמפורים המונים והמשתנים בהם השתמשתם, וכן את ערכי ההתחלה שלהם.

62. (University of Maryland)

עליכם לתזמן את המשימות הבאות. כל המשימות מגיעות בזמן 0 לפי הסדר P_1, \dots, P_4 (המשימה P_1 מגיעה ראשונה) יש לתזמן לפי:

א. FCFS.

ב. SJF.

ג. OPTIMAL.

ד. עדיפויות ללא לקיחה בכוח (מספר עדיפות נמוך יותר מציין עדיפות גבוהה).

ה. עדיפויות עם לקיחה בכוח (מספר עדיפות נמוך יותר מציין עדיפות גבוהה).

ו. ויסות מעגלי ($\text{quantum} = 1$).

כמו כן יש לציין מהו זמן הסבב (turnaround time) עבור כל תהליך בכל שיטה.

| תהליך | זמן ביצוע | עדיפות |
|-------|-----------|--------|
| P_1 | 10 | 1 |
| P_2 | 2 | 3 |
| P_3 | 1 | 4 |
| P_4 | 5 | 2 |

63. (University of Maryland)
לתהליך יש טבלת דפים בגודל 2K כניסות וכמו כן מוקצים לו 4 דפים פסיים. בהתחלה הדפים הפסיים ריקים, והתהליך רוצה לפנות לדפים הבאים (משמאל לימין):
128, 12, 3, 14, 128, 8, 3, 12, 14, 8, 15
- א. הראו את תהליך הקצאת הדפים עבור אלגוריתם LRU. כמו כן יש לציין בסוף את מספר תקלות הדף ואת התוכן הסופי של 4 הדפים הפסיים.
- ב. הראו את תהליך הקצאת הדפים עבור אלגוריתם FIFO. כמו כן יש לציין בסוף את מספר תקלות הדף ואת התוכן הסופי של 4 הדפים הפסיים.
- ג. הראו את תהליך הקצאת הדפים עבור אלגוריתם OPTIMAL. כמו כן יש לציין בסוף את מספר תקלות הדף ואת התוכן הסופי של 4 הדפים הפסיים.
- ד. מהו המספר המינימלי של תקלות דף האפשרי, מדוע?

64. (University of Maryland)
לתהליך יש טבלת דפים בגודל 256 כניסות וכמו כן מוקצים לו 3 דפים פסיים. בהתחלה הדפים הפסיים ריקים, והתהליך רוצה לפנות לדפים הבאים (משמאל לימין):
1, 2, 3, 4, 2, 1, 5, 2
- א. הראו את תהליך הקצאת הדפים עבור אלגוריתם LRU. כמו כן יש לציין בסוף את מספר תקלות הדף ואת התוכן הסופי של 4 הדפים הפסיים.
- ב. הראו את תהליך הקצאת הדפים עבור אלגוריתם FIFO. כמו כן יש לציין בסוף את מספר תקלות הדף ואת התוכן הסופי של 4 הדפים הפסיים.
- ג. הראו את תהליך הקצאת הדפים עבור אלגוריתם OPTIMAL. כמו כן יש לציין בסוף את מספר תקלות הדף ואת התוכן הסופי של 4 הדפים הפסיים.
- ד. מהו המספר המינימלי של תקלות הדף האפשרי, מדוע?

65. (University of Maryland)
להלן זמני הגעה וביצוע של 4 תהליכים (במילי-שניות) שמגיעים לתזמון. זמני הביצוע ידועים מראש למתזמן:

| זמן הגעה | זמן ביצוע | |
|----------|-----------|----|
| 0 | 11 | J1 |
| 2 | 5 | J2 |
| 3 | 9 | J3 |
| 7 | 10 | J4 |

- א. עבור FCFS: קבע את זמן התגובה הממוצע (average response time), המספר המקסימלי של תהליכים במערכת בכל זמן ואת הזמן בו המערכת מסיימת לתזמן את כל התהליכים.
- ב. כנ"ל עבור SJF ללא לקיחה בכוח.
- ג. כנ"ל עבור SJF עם לקיחה בכוח.
- ד. כנ"ל עבור האלגוריתם האופטימלי.

66. (University of Maryland)
לפניך 3 תהליכים שמממשים את אלגוריתם המניעה ההדדית.
- | | | | | | |
|-----|------------------|-----|------------------|-----|------------------|
| PA: | loop | PB: | loop | PC: | loop |
| | entry code | | entry code | | entry code |
| | critical section | | critical section | | critical section |
| | exit code | | exit code | | exit code |
| | end | | end | | end |

הנח כ"י:

- א. exit code ו entry code משתמשות בהמתנה-מעסיקה (למשל, ע"י לולאות test-and-set).

- ב. זמן הביצוע של ה-entry code הוא 0 אם אף תהליך לא נמצא בקטע הקריטי שלו (למשל, ה-lock חופשי).
- ג. זמן הביצוע של critical section, exit code, end הוא 10 מילי-שניות.
- ד. 3 התהליכים רצים על מעבד אחד שמשתמש בתזמון ויסות מעגלי עם $quantum = 5\text{ ms}$.

- מהו התפוקה (throughput), כלומר: מהו מספר הקטעים הקריטיים המבוצעים בשנייה?
- מה ניתן להגיד על זמן ההמתנה של תהליך, כלומר: מה ניתן להגיד על הזמן שלוקח לבצע את ה-entry_code?

67. (University of Maryland)

תוך שימוש בסמפורים, עדכנו את הקוד הבא כך שבכל זמן נתון לכל היותר 3 מהתהליכים P_1, P_2, \dots, P_n מבצעים את ה- T_i שלהם.

```
P1:  loop      P2:  loop      ...      Pn:  loop
      S1;          S2;          ...          Sn;
      T1;          T2;          ...          Tn;
      end          end          ...          end
```

68. (University of Maryland)

יצירת מים דורשת ששני אטומי מימן ואטום חמצן אחד יתחברו. הניחו שהאטומים הם תהליכים ושבידוק 2 תהליכי מימן ותהליך חמצן אחד חייבים לקרוא לפרוצדורה makeWater בזמנית במטרה ליצור מים. השתמשו בסמפורים על-מנת לספק פתרון שאין בו הרעבה והמתנה מעסיקה. הקפידו לציין את ערכי ההתחלה של הסמפורים בהם אתם משתמשים.

69. (University of Maryland)

לתהליך יש טבלת דפים בגודל 256 כניסות וכמו כן מוקצים לו 4 דפים פיסיים. בהתחלה הדפים הפיסיים ריקים, והתהליך רוצה לפנות לדפים הבאים (משמאל לימין):

1, 2, 3, 4, 2, 5, 6, 2, 3, 2, 1, 7, 8

- א. הראו את תהליך הקצאת הדפים עבור אלגוריתם LRU. כמו כן יש לציין בסוף את מספר תקלות הדף ואת התוכן הסופי של 4 הדפים הפיסיים.
- ב. הראו את תהליך הקצאת הדפים עבור אלגוריתם FIFO. כמו כן יש לציין בסוף את מספר תקלות הדף ואת התוכן הסופי של 4 הדפים הפיסיים.
- ג. הראו את תהליך הקצאת הדפים עבור אלגוריתם OPTIMAL. כמו כן יש לציין בסוף את מספר תקלות הדף ואת התוכן הסופי של 4 הדפים הפיסיים.
- ד. מהו המספר המינימלי של תקלות דף האפשרי, מדוע?

70. הפתרון הבא מנסה לממש סמפור כללי ע"י סמפור בינרי. אולם, המימוש אינו נכון. מה הטעות באלגוריתם? כיצד היא נגרמת?

```
TYPE Semaphore IS RECORD =
  val      : INTEGER;
  mutex    : Binary_Semaphore := 1;
  sleep    : Binary_Semaphore := 0;

PROCEDURE P( sem : Semaphore ) =
  P( sem.mutex );
  DEC( sem.val );
  IF sem.val < 0 THEN
    V( sem.mutex );
    P( sem.sleep );
  ELSE
    V( sem.mutex );
  END;
```

```
PROCEDURE V( sem : Semaphore ) =  
  P( sem.mutex );  
  INC( sem.val );  
  IF sem.val <= 0 THEN  
    V( sem.sleep );  
  END;  
  V( sem.mutex );
```

.71

א. כיצד שונים זה מזה האלגוריתמים הבאים לחיפוש בדיסק: FIFO, SSTF, SCAN, C-SCAN.

ב. האם זמן ההמתנה הממוצע זהה עבור כל גילי הדיסק כשמשמשים ב-SSTF?

(University of Waterloo) .72

מהם היתרונות והחסרונות בפלח זמן גדול באלגוריתם תזמון?

(University of Waterloo) .73

א. מערכת ההפעלה תומכת בתזמון עם לקיחה בכוח ופלח זמן של 2. תהליך רץ מופסק רק לאחר שפלח הזמן שלו הסתיים או אם התהליך סיים את ביצועו. מהו זמן התגובה הממוצע וזמן הסיום עבור התהליכים הבאים:

| מספר תהליך | זמן הגעה | זמן ביצוע |
|------------|----------|-----------|
| 1 | 0 | 4 |
| 2 | 1 | 6 |
| 3 | 3 | 4 |

ב. ענו על השאלה כאשר פלח הזמן הוא 3.

ג. ענו על השאלה כאשר אין לקיחה בכוח והאלגוריתם הוא SJF.

(University of Waterloo) .74

במעבד קיימים k תהליכים מוכנים לריצה בזמן 0, ואין תהליכים נוספים. תהליך i ($0 < i \leq k$) דורש i יחידות זמן לביצוע.

א. עבור תזמון round-robin עם לקיחה בכוח ופלח זמן של 1. מהו זמן תגובה ממוצע לתהליכים?

ב. עבור תזמון SJF ללא לקיחה בכוח מהו זמן תגובה ממוצע לתהליכים?

ג. אם לאחד מהמתזמנים הנ"ל יש זמן תגובה ממוצע קטן יותר מהשני, האם זה אומר שהוא מתזמן טוב יותר? מדוע?

(University of Waterloo) .75

ציין לפחות 3 דברים שתהליך רץ יכול לעשות שיגרמו למתזמן לא להעבירו למצב Ready כשהוא מסיים את הריצה.

(University of Waterloo) .76

מה ההבדל בין מתזמן עם לקיחה בכוח לבין מתזמן ללא לקיחה בכוח?

(University of Waterloo) .77

מה ההבדל בין swapping ל-paging?

(University of Waterloo) .78

מה ההבדל בין מקטעים לדפים?

79. (University of Waterloo)
מה זה זיכרון אסוציאטיבי?

80. (University of Waterloo)
לפניך רשימת מאורעות:

| תיוית | מאורע | תיאור |
|-------|---------------------|---|
| S | שגיאת System call | טיפול בשגיאה עקב ביצוע קריאת מערכת. |
| E | שגיאה אחרת | טיפול בשגיאה עקב תקלה בדף, שגיאה אריתמטית וכו'. |
| T | פסיקת תזמון (timer) | טיפול עקב פסיקת תזמון. |
| I | פסיקות אחרות | טיפול בפסיקות אחרות. |

המערכת משתמשת בתזמון round-robin עם לקיחה בכוח. הניחו שתהליך יכול להיות באחד מ-4 מצבים: רץ (running), מוכן (ready), חסום (blocked), סיום (done). מצב סיום מציין שהתהליך סיים את ביצועו. שרטטו דיאגרמת מעברי מצבים כשכל מצב מתאים לכל אחד ממצבי התהליכים הנ"ל. הדיאגרמה צריכה להכיל חץ מכוון ממצב אחד למשנהו, שהתווית שלו היא אחד מ-4 האירועים מהטבלה הנ"ל, אם זה אפשרי שאירוע זה יגרום למעבר מהמצב האחד למצב השני. לדוגמה, צריך להיות חץ מכוון ממצב "רץ" למצב "סיום" עם תווית S, כיוון שזה אפשרי שקריאת מערכת תגרום לתהליך רץ לסיים את ביצועו (למשל, ע"י exit()).

81. (University of Waterloo)

אחד מהפרמטרים של זיכרון וירטואלי הוא גודל הדף. תנו יתרון אחד וחסרון אחד לבחירת גודל דף גדול מאשר גודל דף קטן. נמקו.

82. (University of Waterloo)

מחשב בו לתהליכים יש 1,024 דפים במרחב הזיכרון שלהם, שומר את טבלת הדפים שלו בזיכרון. התקורה לקריאת מילה מטבלת הדפים היא 500 nsec. על-מנת להקטין תקורה זו, למחשב יש זיכרון אסוציאטיבי שמכיל 32 זוגות (דף וירטואלי, מסגרת דף פיסי), וניתן לחפש ב-100 nsec. מהו ה-hit rate הנחוץ כדי להקטין את התקורה הממוצעת ל-200 nsec?

83. (University of Waterloo)

האם אלגוריתם ההזדקנות (aging) עבור החלפת דפים משתמש במחסנית? מדוע?

84. (University of Waterloo)

הניחו שתור בקשות הדיסק מכיל את בקשות למגזרים בגלילים הבאים:
400, 20, 19, 74, 399
הניחו שמיקום זרוע הדיסק נמצא בגליל 200. באיזה סדר יטופלו הבקשות באלגוריתם SSTF? באלגוריתם SCAN? באלגוריתם C-SCAN?

85. (University of Waterloo)

בקשות דיסק מגיעות להתקן הדיסק עבור מסילות: 10, 22, 20, 2, 40, 6, 38 (משמאל לימין).
חיפוש לוקח 5 מילי-שניות עבור כל מסילה. כמה זמן חיפוש נחוץ עבור:
א. FCFS.
ב. Closest track next.
ג. אלגוריתם המעלית - SCAN (תחילה זזים כלפי מעלה).
בכל המקרים הזרוע ממוקמת בהתחלה במסילה 20.

86. (University of Waterloo)

מה זה fragmentation?

87. (University of Waterloo)

הניחו שבדיסק יש סה"כ 100 גלילים.

א. בהנחה שמשתמשים ב-SSTF וזרוע הדיסק ממוקמת בהתחלה מעל גליל 42. תור בקשות הגלילים הוא: 27, 40, 90, 37 (משמאל לימין). מה יהיה המרחק הכולל (בגלילים) שהזרוע תזוז עד שכל הבקשות ימולאו?

ב. אותה שאלה כמו ב-א' כאשר הנתונים כמו ב-א' עבור אלגוריתם המעלית (SCAN) כשבתחילה התזוזה היא כלפי "מעלה", כלומר למספר גלילים גבוהים יותר.

88. בפתרון בעיית הקטע הקריטי, מעבר לתנאי מניעה הדדית עמדנו על התנאים של התקדמות (progress) והמתנה חסומה. מה הקשר/יחס בין 2 תנאים אלו והמושגים קיפאון והרעבה בהתאמה? האם הם הפכים או משלימים האחד של השני בהתאמה? הסבירו.

89. מה הן כל תוצאות הריצה האפשריות של התוכנית הבאה (יש להניח שמספר ה-pid של ה-child הוא 10 ומספר ה-pid של ה-parent הוא 9).

```
int main()
{
    int pid;
    if ( pid = fork() == 0)
    {
        printf("I am = %d, Parent = %d\n", getpid(), getppid());
    }
    else
        printf("Child = %d, I am = %d\n", pid, getpid());
}
```

90. כמה פעמים התוכנית הבאה מדפיסה Hello?

```
#include <stdio.h>
int main()
{
    fork();
    fork();
    fork();
    printf( "Hello\n" );
}
```

91. כמה פעמים התוכנית הבאה מדפיסה Hello?

```
#include <stdio.h>
main()
{
    fork();
    printf("Hello\n");
    fork();
    printf("Hello\n");
    fork();
    printf("Hello\n");
}
```

92. נתונה התוכנית הבאה:

```
#include <stdio.h>
main()
{
    int pid;
    printf("ORIGINAL: PID=%d PPID=%d\n", getpid(),
getppid());
    pid = fork();
    if (pid)
        printf("PARENT: PID=%d PPID=%d pid=%d\n",
            getpid(), getppid(),
pid);
    else
        printf("CHILD: PID=%d PPID=%d pid=%d\n",
            getpid(), getppid(),
pid);
}
```

הפלט יראה בצורה הבאה:

```
ORIGINAL: PID=A PPID=B
CHILD: PID=C PPID=D pid=E
PARENT: PID=F PPID=G pid=H
```

כאשר A-H מייצגים מספרים. בהנחה ש-A = 99, B = 42, תנו דוגמה (אחת) למספרים ש-C-H יכולים לייצג.

93. לאיזו מההוראות הבאות צריך לאפשר להתבצע במצב משגוח (monitor mode) בלבד?
א. קריאת שעון זמן של המערכת.
ב. קביעת שעון זמן.
ג. אישור (enable) פסיקת קוצב זמן של המערכת.
ד. חסימת פסיקת קוצב זמן.
ה. העברה למצב משגוח.

94. במערכת VAX\VMS יש חלוקה ל-4 מצבי מערכת שונים:

User, Supervisor, Executive, Kernel

כאשר מצב גרעין (Kernel) הוא הכי מיוחס (privileged) ואילו מצב משתמש (User) הוא הכי לא מיוחס. מדוע יש צורך ביותר מאשר שני מצבי מערכת (גרעין ומשתמש)? תנו דוגמאות ספציפיות לסוגי מערכות/תוכנות שהגיוני להריץ בכל אחד מארבעת מצבי המערכת השונים.

95. ברוב מערכות ההפעלה יש מושג של משתמש-על (super user) או root. למה תפקיד זה נחוץ? מה הבעיות שתפקיד זה יכול ליצור במערכת? הסבירו.

96. הגדרה אחת של מערכת הפעלה היא אוסף של אלגוריתמים המסתירים את המורכבות והפרטים של החומרה כדי לספק למשתמש סביבה נעימה יותר. איך מתקשרת הגדרה זו עם ההגדרה שניתנה בכיתה של מערכת ההפעלה כמכונה בפועל?

מילון מונחים:

thread - תהליכון -
deadlock - קיפאון -
round robin - ויסות מעגלי -
segment - מקטע -
task = משימה = תהליך עם תהליכונים =
preemption = לקיחה בכוח = שחרור לא רצוני =
page fault = תקלות דף =
fragmentation = פיצול =
frame = מסגרת =
simulate = הדמה ("סמלץ")

מערכות הפעלה 88-288

תרגיל בית תיאורטי

מספר 1

מגיש/ה:

שם מלא: _____

ת.ז./דרכון:

| | | | | | | | | | |
|--|--|--|--|--|--|--|--|---|--|
| | | | | | | | | - | |
|--|--|--|--|--|--|--|--|---|--|

מספר קבוצה:

| | | | | | | | | |
|---|---|---|---|---|---|---|--|--|
| 8 | 8 | - | 2 | 8 | 8 | - | | |
|---|---|---|---|---|---|---|--|--|

/100

ציון

מערכות הפעלה 88-288

תרגיל בית תיאורטי

מספר 2

מגיש/ה:

שם מלא: _____

ת.ז./דרכון:

| | | | | | | | | | |
|--|--|--|--|--|--|--|--|---|--|
| | | | | | | | | - | |
|--|--|--|--|--|--|--|--|---|--|

מספר קבוצה:

| | | | | | | | | |
|---|---|---|---|---|---|---|--|--|
| 8 | 8 | - | 2 | 8 | 8 | - | | |
|---|---|---|---|---|---|---|--|--|

/100

ציון

מערכות הפעלה 88-288

תרגיל בית תיאורטי

מספר 3

מגיש/ה:

שם מלא: _____

ת.ז./דרכון:

| | | | | | | | | | |
|--|--|--|--|--|--|--|--|---|--|
| | | | | | | | | - | |
|--|--|--|--|--|--|--|--|---|--|

מספר קבוצה:

| | | | | | | | | |
|---|---|---|---|---|---|---|--|--|
| 8 | 8 | - | 2 | 8 | 8 | - | | |
|---|---|---|---|---|---|---|--|--|

/100

ציון

הנחיות כלליות בנוגע לתרגילים:

- מדיניות איחורים:
תרדנה 0.5 נקודות עבור כל שעת איחור עד 24 שעות.
החל מהשעה ה-25 ועד 48 שעות איחור: תרדנה 2 נקודות עבור כל שעת איחור.
לאחר 48 שעות לא תתאפשר ההגשה והציון יהיה 0.
לדוגמה, הציון המקסימלי עבור סטודנט/ית שהגיש/ה תרגיל 29 שעות לאחר המועד יהיה 78.

- מדיניות העתקות:
תרגיל שייחשד כמועתק יקבל ציון 0. עבור תרגיל עבורו יש חובת הגשה על-מנת לעבור את הקורס, הדבר יגרוור כשלון בקורס.

נושאי האיחורים וההעתקות יאכפו כמה שניתן ולא תהיינה התפשרויות בנושאים אלו.

אין לפנות למתרגלים בעקבות הורדת ניקוד עקב איחורים. אם ידועה סיבה מוצדקת לאיחור יש לפנות לפני הגשת התרגיל ולקבל אישור מהמתרגל.

- הגשה:
העמוד הראשון יהיה "עמוד שער". לכל תרגיל יצורף עמוד שער אותו יש למלא בכתב ברור. **חובה** שהעמוד הראשון בכל תרגיל יהיה עמוד שער כמצורף לתרגיל. תרגילים שלא יכללו עמוד כנ"ל לא יבדקו.

על התרגילים להיות **מודפסים** וכתובים **בעברית**. תרגילים לא מודפסים לא יבדקו.

על סטודנטים המתקשים בעברית לפנות **לפני** הגשת התרגיל למתרגלים ולבקש אישור להגיש **באנגלית**.