

בחיינה במערכות הפעלה (88-288), סמסטר ב' מועד א'

מס' מחברות:

- משך הבחינה: שעותיים.
 - הבחינה נערכת בחומר סגור וראש פתוח.
 - הבחינה כוללת 12 שאלות. כל שאלה דמוי-אמריקאית (דמוי-ישראלית) מקנה לפותר נכונה עד 6 (10) נקודות.
 - סמנו כל תשובה על גבי טופס זה ע"י עיגול סביב האות המזהה בצורה חד-משמעית. אפשרי ללוות את התשובה בהסבר קצר במחברת.
 - בשאלה דמוי-אמריקאית יש לסמן את האותיות של כל התשובות הנכונות לשאלה.
 - השתמשו במחברת כטיוטה ולצורך תשובות קצרות על השאלות דמוי-ישראליות.
 - באם נראה לכם שתשובתכם תלויה ישירות בהנחה מסוימת, רשמו אותה בתשובה לשאלה במחברת.
 - בכל שאלה יש נתונים מספיקים בכדי לענות עליה.
 - בתום הבחינה מסרו את מחברת הבחינה ואת טופס הבחינה מצורפים יחד.
- בהצלחה!

1. בין ההבדלים בין מכלאים (buffering) ומעקפים (spooling) ניתן (אולי) למנות:

- (1) מכלאים עושים שימוש בטכניקת DMA אבל מעקפים לא.
- (2) מכלאים מאפשרים חפיפה בין העיבוד של תכנית (תהליך) למטלות הק/פ שלה, אבל מעקפים לא.
- (3) מעקפים מאפשרים חפיפה בין העיבוד של תכנית למטלות הק/פ של תכניות אחרות, אבל מכלאים לא.
- (4) כמו במכלאים, אפשר להשתמש ברעיון של מעקפים רק בין הדיסק לזיכרון העיקרי.

אילו מהטיועונים לעיל נכונים?

א. 1

ב. 2

ג. 3

ד. 4

ה. אף אחד מהטיועונים.

2. בפתרון בעיית הקטע הקריטי, מעבר לתנאי המניעה ההדדית (mutual exclusion) עמדנו על התנאים של התקדמות (progress) והמתנה חסומה (bounded wait). מה הקשר/יחס בין שני תנאים אילו והמושגים קיפאון (deadlock) והרעבה (starvation) בהתאמה? האם הם הפכים או משלימים האחד של השני בהתאמה? הסבירו.

3. בין הטיועונים לגבי סמפורים (semaphores) בכלל ניתן (אולי) למנות:

- (1) שתי פעולות P (wait) המופעלות על סמפורים שונים חייבים להיות מנועים הדדית.
- (2) הערך התחילי של סמפור לא יכול להיות שלילי.
- (3) מכניזם התור של סמפור חייב להיות בשיטת FIFO (נכנס ראשון יוצא ראשון).
- (4) אין טעם לספק פעולות תיחול (initialize) וקריאת ערך של סמפור כפעולות בסיסיות של סמפור.

אילו מהטיועונים לעיל נכונים?

א. 1

ב. 2

ג. 3

ד. 4

ה. אף אחד מהטיועונים.

ישנם הרבה דרכים לעדכון פלח הזמן q המוקצה לתהליך מסוים בכל אדגם (instance) שלו באלגוריתם ויסות מעבד RR. בין האפשרויות:

- (1) q גודל באחד אחרי כל אדגם.
- (2) כמו ב-1, אבל q נשאר קבוע אחרי הגעתו לערך Q_{max} .
- (3) ה- q המוקצה לכל תהליך הוא בצורה $q, 2q, 4q, 8q, q, 2q, 4q, 8q, \dots$

מהו הרעיון מאחורי העדכון של q ב-RR. מהו הרעיון הספציפי מאחורי כל אחת משלוש אפשרויות אלו? הסבירו.

5. בין הטיעונים לגבי מודל התהליכים ולגבי מודל המשימה פלוס תהליכונים אפשר (אולי) למנות:

- (1) פלח הזמן בין מיתוגי הקשר תהליכונים יכול להיות קצר יותר מאשר פלח הזמן הכדאי בין מיתוגי הקשר תהליכונים.
- (2) תקלת דף הקוראת לתהליכון כלשהו במשימה בהכרח משביתה את כל התהליכונים במשימה עד להתגברות על תקלת הדף.
- (3) במודל המשימה פלוס תהליכונים יש טעם לווסת ברמת המשימות מעבר לויסות הנדרש ברמת התהליכונים.
- (4) קל יותר לבצע הדמיה (simulation) למודל התהליכים במסגרת מודל המשימה פלוס תהליכונים מאשר להפך.

אילו מהטיעונים לעיל נכונים?

- א. 1
- ב. 2
- ג. 3
- ד. 4

ה. אף אחד מהטיעונים.

6. נתונה מערכת זיכרון וירטואלי עם דפדוף. הזיכרון הפיסי מכיל 8 מסגרות, כל אחת בגודל 4 בתים. נתונה תוכנית בעלת 12 דפים לוגיים. אילו מהמיפויים הבאים, בין כתובת לוגית לכתובת פיזית (יכולים להיות) חוקיים (התייחסו לכל מיפוי בנפרד) בהקשר הנתון? (הערה: כל המספרים הנתונים כאן הם דצימליים).

כתובת פיזית	כתובת לוגית
א. 31	→ 27
ב. 17	→ 57
ג. 1	→ 29
ד. 37	→ 41
ה. 12	→ 14

7. במערכת זיכרון וירטואלי עם דפדוף נתונה מחרוזת ההתייחסות לדפים (page reference string) הבאה (משמאל לימין):

1,2,3,4,5,3,4,1,6,7,8,7,8,9,7,8,9,5,4,5,4,2

מהו מספר תקלות הדף ומספר החלפות הדף שיקרו לאלגוריתמי ההחלפה של OPT, FIFO ו-LRU בהנחה של 4 מסגרות בזיכרון הפיסי (שריקות מלכתחילה).



אוניברסיטת בר-אילן המחלקה למתמטיקה ולמדעי המחשב

בחינה במערכות הפעלה (88-288), סמסטר ב', מועד צי' תשנ"ו

נבחן שימצאו ברשותו חומי
עזר אסורים או יתפס בהעתק
יענש בחומרה עד כדי הוצאה
מהאוניברסיטה.

מס' מחברת: _____

- משך הבחינה: שעה
- הבחינה נערכת בחומר סגור (חוץ מדף סמלים) וראש פתוח.
- הבחינה כוללת 12 שאלות. כל שאלה דמוי-אמריקאית (דמוי-ישראלית) מקנה לפותר נכונה עד 6 (10 נקודות).
- סמנו כל תשובה על גבי טופס זה ע"י עיגול סביב האות המזהה בצורה חד-משמעית. אפשרי ללוות את התשובה בהסבר קצר במחברת.
- בשאלה דמוי-אמריקאית יש לסמן את האותיות של כל התשובות הנכונות לשאלה.
- השתמשו במחברת כטיטסה ולצורך תשובות קצרות על השאלות דמוי-ישראליות.
- באם נראה לכם שתשובתכם תלויה ישירות בהנחה מסוימת, רשמו אותה בתשובה לשאלה במחברת.
- בכל שאלה יש נתונים מספיקים בכדי לענות עליה.
- בתום הבחינה מסרו את מחברת הבחינה ואת טופס הבחינה מצורפים יחד.

בהצלחה!

1. ישנם מספר שיטות להעברת ארגומנטים מהשגרה המוזמנת (caller) לשגרה המוזמנת (callee). איזה מהשיטות הבאות ישימה בזימון שגרה כניסתית (reentrant procedure) (ז.א. מספר תהליכים משתמשים בה בו-זמנית) כמו לדוגמא קריאת מערכת (system call).

- א. העברת ארגומנטים באוגרים.
- ב. העברת ארגומנטים בשטח המחסנית של התהליך.
- ג. ארגומנטים מושמים בשטח מוקצה מראש מיד אחרי הקוד של קריאת הזמון.
- ד. ארגומנטים מושמים בשטח מוקצה מראש בראשית השגרה המוזמנת.
- ה. ארגומנטים מושמים בשטח המוקצה דינמית במקטע הנתונים (data segment) של התהליך.

2. מהם ההשלכות של עקרון $0, 1, \infty$ לגבי מקביליות (concurrency) בכלל ולגבי סמפורים (בינאריים וכלליים) בפרט. הסבירו.

3. במקביליות, בין הטיעונים לגבי הפתרון (המוצג בהמשך) לבעיית היצרן/צרכן (Producer/Consumer) תוך שמוש בסמפורים (המוצג בהמשך) אפשר (אולי) למנות:

- (1) הפתרון נכון רק ליצרן אחד ולצרכן אחד ולא למספר יצרנים/צרכנים במקביל.
- (2) סכום הסמפורים space ו-items בכל שלב התחלתי של LOOP הוא N.
- (3) הפעולות על הסמפורים space ו-items מנועים הדדית (mutually exclusive).
- (4) יתכן מצב באלגוריתם שבו יש פריט במאגר אבל הצרכן עדיין ממתין על הסמפור items.

VAR mutex : semaphore; {initialise to 1}
space : semaphore; {initialise to N}
items : semaphore; {initialise to 0}

אילו מהטיעונים לעיל נכונים?

PROCESS Producer;
LOOP
- produce item;
P(space); {wait for space available}
P(mutex); {get mutual exclusion}
- deposit item;
V(mutex); {release mutual exclusion}
V(items); {signal items available}
END.

אף אחד מטיעונים אילו.

PROCESS Consumer;
LOOP
P(items); {wait for items available}
P(mutex); {get mutual exclusion}
- fetch item;
V(mutex); {release mutual exclusion}
V(space); {signal space available}
- consume item;
END.

63

4. אלגוריתמי ויסות מעבד רצוניים (nonpreemptive) כמו FCFS ו-SJF גורמים לאפליות שונות לסוגי תהליכים שונים. מוצע לאזן את האפליות השונות ע"י חישוב עדיפות לכל תהליך לפי הנוסחה

$$\text{פרץ עיבוד} / (\text{זמן המתנה} + \text{פרץ עיבוד}) = \text{עדיפות}$$

כאשר העדיפות ניתנת לתהליך עם המספר הגבוה ביותר. מהם היתרונות והחסרונות של אלגוריתם עדיפות זה לעומת FCFS ו-SJF? איך הוא משיג את האיזון בין האפליות שהוא משיג?

השוו ע"י תרשים גאנט וחישוב ממוצע זמן סבב (או המתנה) בין האלגוריתמים SJF, FCFS ואלגוריתם העדיפות המוצע בדוגמא:

תהליך	זמן הגעה	פרץ עיבוד
A	0	3
B	2	5
C	3	2
D	5	5
E	7	2

5. במערכת זכרון וירטואלי עם קטוע (segmentation) יש תהליך A. כתובת לוגית מורכבת משלוש ספרות דצימליות, כאשר ספרת המאות היא מספר המקטע ושתי הספרות הנותרות הן ההיסט (בבתים, מ-0 ומעלה) במקטע. טבלת הסגמנטים של A היא:

מספר סגמנט	התחלה	גודל (בבתים)	הרשאות גישה
0	13426	26	RWX
1	2309	0	R
2	1000	23	WX
3	0	95	W
4	10000	100	RWX

A מבצע סדרה של ניסיונות גישה לזכרון (מבלי להיזרק בעת כישלון) לפי הכתובות (מלמעלה למטה):

כתובות לוגית	סוג גישה
000	R
100	R
217	W
217	R
223	W
232	R
394	R
499	X
725	X

כמה מניסיונות הגישה של A נכשלות (מסיבה כלשהי)?

- א. 5
ב. 6
ג. 8
ד. 9
ה. אף אחד ממספרים אילו.

64

6. אחת מהבעיות בניהול מדרג של תהליכים הוא סיום תהליך. לגבי תהליך הורה (parent) שיצר (fork/spawn) תהליכי ילד (child), ישנם שלוש אפשרויות:

- א) אסור להורה לסיים או להיהרג (kill) לפני סיום או הרג מפורש של ילדיו.
- ב) סיום/הרג הורה פירושו גם הרג אוטומאטי של כל ילדיו (cascading termination).
- ג) ההורה מסיים/נהרג, אך ילדיו ממשיכים לרוץ.

יהשוו (במבנה טבלה) בין שלוש אפשרויות אלו. מהי גישת מערכת ההפעלה UNIX לבעייה זו? הסבירו.

7. בזכרון וירטואלי, נתונה מחרוזת ההתייחסות לדפים (page reference string) הבאה (משמאל לימין):

3 2 1 0 3 2 4 3 2 1 0 4

שרטטו את גרף תקלות הדפים (מספר תקלות דף לעומת מספר מסגרות פיסיות) בהנחה של 1-6 מסגרות פיסיות בשמוש אלגוריתם LRU.

8. ישנם שיקולים לכאן ולכאן לגבי הגדלת גודל דף (page size) במערכת זכרון וירטואלי עם דפדוף. אילו מהשיקולים הבאים מהווים אילוץ להגדלת גודל הדף?

- א. גודל טבלת הדפים (מספר הכניסות).
- ב. עלות ק/פ לדף.
- ג. פיצול פנימי.
- ד. פיצול חיצוני.
- ה. עקרון המקומיות.

9. שיטת הקצאת דיסק הנהוגה בעולם ה-IBM הוא שיטת התוספות (Extents). מעבר להקצאת הרציפה הראשונית הגדולה יש אפשרות לבקש עד כ-15 תוספות הקצאה נפרדות של שטחים רציפים (לאו דוקא ברציפות לשטחים שהוקצו קודם). כמוכן שצריך אינדקס כדי לזכור את הכתובות של כל השטחים המוקצים לקובץ. שיטת התוספות היא איזשהו שילוב של הקצאה רציפה (contiguous allocation) והקצאה אינדקס (indexed allocation). מהם היתרונות והחסרונות (בצורה טבלה) של שיטת התוספות לעומת שתי שיטות ההקצאה האלו? הסבירו.

10. בויסות דיסק, בהתייחס לשלושת האלגוריתמים של FCFS, SSDF ו-LOOK, ישנם שש תמורות (פרמוטציות) אפשריות של זוגות ביניהם (3.2). לגבי כל תמורה, הראו דוגמא כלשהי של סדרה של שלוש בקשות ק/פ (לפי גלילים), כאשר לאלגוריתם הראשון יש מרחק חיפוש כולל קטן יותר מאשר לשני בוג. הניחו כל מה שדרוש.

65

11. בין הטיעונים לגבי מנגנון האיתות (signals) ב-UNIX אפשר (אולי) למנות:

- (1) קריאות המערכת (system calls) של שקעים (sockets) מיושמים על ידי שימוש באיתות.
- (2) מלכודות (traps) ב-UNIX מיושמים על ידי שימוש באיתות.
- (3) יש הבדל בין איתות SIGSTOP (17) לבין איתות SIGKILL (9).
- (4) אין בכלל קשר בין הקשר האיתות (signalling context) של תהליך הילד ושל תהליך ההורה שלו.

אילו מהטיעונים לעיל נכונים?

- א. 1
- ב. 2
- ג. 3
- ד. 4
- ה. אף אחד מטיעונים אילו.

12. כאשר מורצת התכנית הבאה, מה התוצאות? הסבירו.

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
main ()
{
    FILE *a;
    char s[10];

    unlink("a");
    mknod("a", S_IFIFO | S_IRWXU, 0);
    a = fopen("a", "w");
    fprintf(a, "a");
    fclose(a);
    a = fopen("a", "r");
    fscanf(a, "%s", &s);
    printf(s);
    fclose(a);
}
```

76
1120
121
P 121
25/9/97

3 7 1 3

אוניברסיטת בר-אילן
המחלקה למתמטיקה ולמדעי המחשב

בחינה בקורס מערכות הפעלה

סמסטר ב' מועד ב', תשנ"ז

מס' קורס: 88-288

שמות המרצים: ד"ר אריאל פרנק, ד"ר גלעד קונרד, ד"ר אורלי קדמין
תאריך הבחינה: 25/9/97
זמן: שעתיים

- אם נראה לכם שתשובתכם תלויה ישירות בהנחה מסוימת, רשמו אותה בתשובה לשאלה במחברת.
- בכל שאלה ישנם נתונים מספיקים לענות עליה.
- יש לענות על ארבע מתוך חמש השאלות
- חומר עזר אסור

שאלה 1:

א. השוו בין השימוש בסמפור ובין השימוש ב- TS (Test and Set) להגנה על קטע קריטי. יש מערכות הפעלה שתומכות בשני המנגנונים גם יחד. האם יש בכך הגיון או בזבוז? הסבירו.

ב. נניח שבמערכת מסוימת יש פקודה אטומית שנקראת TAA (Test and Add) המופעלת על משתנה. אם ערכו של המשתנה הוא אפס, פקודה זו מעלה את ערכו של המשתנה באחד, אחרת, אין שינוי בערכו של המשתנה. בנוסף, מחזירה הפקודה אינדיקציה האם המשתנה המסוים היה מאופס לפני הפעולה או לא.

לדוגמא, בפקודה: `count, if_zero = TAA count` אם `count` היה בתחילה מאופס, אזי הפקודה תעלה את ערכו של המשתנה ב-1. בנוסף, ערכו של המשתנה `if_zero` יהיה TRUE, אחרת, `count` לא ישתנה וערכו של המשתנה `if_zero` יהיה FALSE.

במערכת עם N תהליכים, לכל תהליך יש `id` יחיד כמספר בין אפס ל- `N-1`. מוצע הפתרון הבא לבעיית הקטע קריטי:

```
TEST: TAA count, if_zero
      turn = (turn + 1) MOD (N - 1)
      if (NOT if_zero) {
          go to TEST
      }
      else {
          if (my_id < turn) {
              count = count - 1
              go to TEST
          }
          else {
              access other shared data /* the critical section */
              count = count - 1
          }
      }
```

כאשר `count` ו-`turn` הם משתנים משותפים ו-`if_zero` הוא משתנה מקומי. `my-id` מכיל לכל תהליך את המספר המזהה שלו.

האם ניתן בעזרת האלגוריתם הנ"ל לפתור את בעיית הקטע קריטי? באילו הנחות? (הראו מדוע ההנחות שהנחתם, אם בכלל, חיוניות).

לגבי כל אחת משלוש הדרישות לקיום פתרון, הוכיחו את קיומה או לחליפין הראו דוגמא המפירה אותה.

מסמך מס' 78
 מס' 78
 מס' 78
 מס' 78

שאלה 2:

- א. מה יכול להיות הגורם להמתנה אינסופית של תהליך בתור המוכנים (ready queue)? (יש להניח שכל התהליכים הם בלתי תלויים). האם התנהגות כזו אפשרית כאשר משתמשים ב- FCFS או RR (Round Robin) לתזמון? נמקו.
- ב. הביאו דוגמא לאלגוריתמים תזמון העלול להיות חשוף לתופעה כזו, הסבירו את סיבתה והציעו פתרון לבעיה.
- ג. מתזמן עובד בשיטת RR (מנהל תור ממתנינים בו מופיע כל תהליך מוכן לריצה פעם אחת בלבד). מה יקרה אם תהליך מסוים יופיע פעמיים ברשימה? האם תוכלו למצוא מקרים בהם יש שמוש להופעה כפולה שכזו (מה לגבי משולשת וכו')? הסבירו.

שאלה 3:

- א. נתונה תמונת זכרון המכילה כרגע קטעי זכרון פנויים (חורים) בגדלים הבאים (לפי סדר הופעתם בזכרון): 100K, 200K, 100K, 512K (משמאל לימין).

מה תהיה תמונת הזכרון (החורים) לאחר הבקשות הבאות (משמאל לימין) 90K, 60K, 10K אם הזכרון מנוהל בשיטות:

FIRST FIT (תואם ראשון)
 NEXT FIT (התואם הבא)
 BEST FIT (התואם הטוב ביותר)
 WORST FIT (התואם הגרוע ביותר)

הערה: בכל השיטות הנ"ל אם נוצל חלק משטח פנוי יתרת השטח עדיין פנויה.

- ב. מדוע קל יותר לתמוך בהגנה (Protection) ובשיתוף (Sharing) כאשר הזכרון הוירטואלי מנוהל בשיטת מקטעים (segmentation) מאשר בדפדוף (paging)? הסבירו.

שאלה 4:

נתונה מערכת שבה לכל תהליך יש זכרון וירטואלי בגודל 1024 דפים. כל דף גודלו 4K. כל טבלת הדפים נמצאת בזכרון הראשי. זמן הגישה לטבלת הדפים הוא 500ns (500 ננו שנייה).

כדי להקטין את התקורה הנובעת מגישות לטבלת הדפים, משתמשים בזכרון אסוציאטיבי (טבלת דפים צדדית TLB), בעל 32 כניסות. עלות הגישה ל-TLB היא 100 ns בלבד.

- א. בהקשר זה, הגדירו "יחס פגיעה" (Hit Ratio).
- ב. בדוגמא הנ"ל, חשבו מהו יחס הפגיעה הנדרש כדי להוריד את זמן הגישה הממוצעת הממוצעת ל-200ns.
- ג. כיצד משתלב ה-TLB במערכת MULTIPROGRAMING? האם מתוג הקשר יכול להשפיע על "יחס פגיעה"? אם כן, כיצד? הסבירו.

אוניברסיטת בר-אילן
המחלקה למתמטיקה ולמדעי המחשב

שאלון סגור

בחינה בקורס מערכות הפעלה
סמסטר ב' מועד א', תשנ"ח
מס' קורס: 88-288

שמות המרצים: ד"ר אריאל פרנק, ד"ר גלעד קורן, ד"ר אורלי קרמין
תאריך הבחינה: 9/7/98
זמן: שתיים

- אם נראה לכם שתשובתכם תלויה ישירות בהנחה מסוימת, רשמו אותה בתשובה לשאלה במחברת.
- בכל שאלה ישנם נתונים מספיקים לענות עליה.
- יש לענות על ארבע מתוך חמש השאלות.
- חומר עזר אסור

1.

(א) הסבירו מהם המכניזמים של שחלוף (Swapping) ושל כיווץ (Compaction). איך יכול שחלוף להוריד את זרישות העיבוד הנדרשות לצורך ביצוע כיווץ? מהי החשיבות (אם בכלל) של היכולת לבצע מיקום מחדש (relocation) במערכת התומכת בשחלוף ו/או כיווץ?

(ב) נתונה מערכת שבה לכל תהליך יש זיכרון וירטואלי בגודל 1024 דפים. כל דף גודלו 4K. זמן הגישה לטבלת הדפים הוא 600ns (600 ננו שניה). כדי להקטין את התקורה הנובעת מגישות לטבלת הדפים, משתמשים בזיכרון אסוציאטיבי (טבלת דפים צדדית TLB). בעל 64 כניסות. זמן הגישה ל-TLB הוא 110ns בלבד.

(i) בהקשר זה, הגדירו "יחס פגיעה" (Hit Ratio).

(ii) במערכת מסוימת יחס הפגיעה הוא h. ברשותכם 1000\$ שניתן בעזרתם לבצע את אחד מהשיפורים הבאים:

- (א) לשפר זמן גישה לטבלת הדפים: 500ns בלבד.
- (ב) לשפר זמן גישה ל-TLB: 100ns בלבד.

באיזה מקרים שפור (א) עדיף ובאיזה תנאים שפור (ב) עדיף (במדויק)?

2.

(א) הגדירו מהו סמפור (בינארי וכללי). סמפורים משמשים למניעה הודדית, לסינכרוניזציה (תאום) ולתקשורת. תארו והדגמו בקצור כל אחד משימושים אלה!

(ב) 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 היצור בלבד.
כלומר תוכנית היצור היא:

```

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

```

האם התכנית המתקבלת עדיין פותרת את הבעיה? נמקו!

(ג) הפתרון לבעיית היצרן-צרכן המוצג למעלה (הפתרון המקורי לפני השנוי) ניתן לשימוש גם למקרה ש:

- (i) יש צרכנים רבים אך לא יותר מיצרן אחד.
- (ii) יש יצרנים רבים אך לא יותר מצרכן אחד.
- (iii) יש יצרנים רבים וגם צרכנים רבים אך במספר שווה.
- (iv) יש יצרנים רבים וגם צרכנים רבים (במספר כלשהו).

בחרו את התשובה המתאימה ביותר, ונמקו בקיצור.

3.

(א) באלגוריתם ויסות מעבד SJF (התהליך הקצר קודם) משתמשים בנוסחה שלמדנו להערכת פרץ עיבוד עם $\alpha = \frac{1}{2}$. לתהליך מסוים יש את הסדרה הבאה של פרצי עבוד של הדגמי (instances) התהליך (משמאל לימין): 20,25,20,40

ובשלב הבא, הערכת הפרץ משתווה לאורך פרץ העיבוד בפועל. שניהם שווים ל-30 יחידות זמן. מה הייתה הערכת פרץ העיבוד הראשונית (לפני תחילת הריצה).

(ב)

- (i) מה "המשמעות" הערך של הקבוע α שלמעלה (מהן הסיבות והאילוצים לקביעת α קטן יותר או α גדול יותר)?
- (ii) האם יש מקום לקבוע ערך α שונה לתהליכים שונים?
- (iii) יש הצעה להפוך את הקבוע α הנ"ל למשתנה דינמי שיכול להתעדכן מחדש להדגם. האם יש הגיון בהצעה זו? אם כן, מהו?

בעיבוד מרובב (רב-תכנות multiprogramming) מועברת השליטה על המעבד בין תהליכים שונים באמצעות המתחמן/מוסתר/משגר (Dispatcher/Scheduler) זוהי כמובן תקורה (overhead) של המעבד. כאשר כמה תהליכים נמצאים בעיבוד, מקבל כל תהליך בביצוע פחות זיכרון. זוהי תקורה של הזיכרון וניהולו. תהליכים מתנגשים בבקשות ק/פ (I/O) -- שוב תקורה. למה אם כן טוב (האם באמת טוב) העיבוד המרובב?

(א)

במערכת התומכת בדפדוף לפי דרישה (Demand Paging), מדידות של ניצולת (Utilization) של משאבים מראות:

(ב)

- ניצולת מעבד: 20%
- ניצולת דיסק דפדוף: 99.7%
- ניצולת התקני ק/פ (I/O) : 5%

לגבי כל אחד מהשינויים המוצעים למטה, ענו בבירור האם הם ישפרו את ניצולת המעבד או לא. נמקו תשובותיכם.

- (1) מעבד מהיר יותר.
- (2) העלאת רמת העיבוד המרובב (degree of multiprogramming).
- (3) הורדת רמת העיבוד המרובב.
- (4) דיסק דפדוף גדול יותר.
- (5) התקני ק/פ אחרים, מהירים יותר.
- (6) הגדלת הזיכרון הפנימי.
- (7) הגדלת גודל הדף.

תהליכים יכולים להתקשר ע"י כתיבה לקובץ רגיל וקריאה ממנו. כמו כן תהליכים יכולים להתקשר ע"י כתיבה לקובץ מסוג pipe וקריאה ממנו. מה ההבדל? מה עדיף? האם תמיד עדיף (מתי כן ומתי לא)?

(א)

מה הפלט של התכנית הבאה? נמקו!

(ב)

```
#include <stdio.h>
```

```
main()
```

```
{
int pfd[2],s;
char buf[BUFSIZ];
```

```
pipe(pfd);
if ( fork()!=0 )
{
wait(&s);
close ( pfd[0] );
write ( pfd[1], "Here is some information about son's
work", BUFSIZ );
write ( pfd[1], "This is some additional information.",
BUFSIZ );
close ( pfd[1] );
}
else
{
close ( pfd[1] );
close ( pfd[0] );
}
}
```

אוניברסיטת בר-אילן
המחלקה למתמטיקה ולמדעי המחשב

שאלון סגור

בחינה בקורס מערכות הפעלה
סמסטר ב' מועד ב', תשנ"ח
מס' קורס: 88-288

שמות המרצים: ד"ר אריאל פרנק, ד"ר גלעד קורן, ד"ר אורלי קרמין
תאריך הבחינה: 1/9/98
זמן: שתיים

- אם נראה לכם שתשובתכם תלויה ישירות בהנחה מסוימת, רשמו אותה בתשובה לשאלה במחברת.
- בכל שאלה ישנם נתונים מספיקים לענות עליה.
- יש לענות על ארבע מתוך חמש השאלות.
- חומר עזר אסור.

1. (א) מהי טבלת דפים צדדית (cache/TLB/Look Aside Buffer)? למה היא משמשת? מה המערכת עושה כאשר ה-TLB מלא וצריך להכניס מיפוי נוסף?
- (ב) נתונה מערכת זיכרון בגודל 2MB המנוהלת בשיטת החברים (העמיתים BUDDY SYSTEM). הראו (רצוי לצייר) את מפת הזיכרון (קטעים תפוסים ופנויים בכל שלב) במהלך סדרת ההקצאות והשחרורים הבאה, כאשר בתחילה הזיכרון כולו פנוי:

מוקצה	400K
מוקצה	200K
מוקצה	300K
מוקצה	100K
שלמעלה משוחרר	200K
שלמעלה משוחרר	100K

- (ג) האם שיטת החברים (העמיתים) טובלת מריסוק (פיצול) פנימי? ומריסוק (פיצול) חיצוני (INTERNAL/EXTERNAL FRAGMENTATION)? (נמקו!)

2. (א) האם ניתן לממש סמפור ע"י שימוש בהוראת TS? אם כן, הראו כיצד והעריכו את היתרונות והחסרונות של יישום שכזה לעומת יישומים אפשריים אחרים! אם לא, נמקו מדוע לא?
- (ב) במערכת מחשב מסוימת לא כלולה הוראת TS. לעומת זאת קיימת הוראת SWAP R, M המחליפה באופן אטומי את התוכן של R ו-M. מה פרוש "אטומי" כאן? האם ניתן להשתמש בהוראה זו לצורך מימוש מניעה הדדית? נמקו.

3. (א) בין הטענות לגבי שיטת התזמון RR (Round Robin) ניתן אולי לטעון (לכל אחד מהנ"ל רשום נכון או לא נכון ונמק בשורה או שתיים):
- תזמון RR (לעומת שיטות אחרות כמו תזמון עדיפויות ו-FIFO) יכול:
- (1) לשפר תגובה לתהליכים עתירי ק/I/O.
 - (2) לקצר זמני תגובה בתהליכים אינטרקטיביים.
 - (3) לשפר זמני סיום של תהליכי אצווה (batch).
 - (4) להקטין מספר מיתוגי הקשר.
 - (5) להגדיל את ניצולת המעבד (CPU).

- (ב) מדידות במערכת מסוימת הראו ש:
- אם ניתן לתהליך לרוץ ללא הפרעה, התהליך ירוץ במוצטע במשך זמן T לפני שהוא נכנס למצב ממתין (waiting) (ומחכה למשל לסיום I/O). נתון זה נכון לכל התהליכים במערכת.
 - במערכת זו, מיתוג הקשר (context switch) דורש זמן S (שהוא למעשה זמן תקורה (overhead) שמבזבז).
- אנו שוקלים לתזמן מערכת זו ע"י RR עם פלח זמן Q כלשהו.

לכל אחד מהמקרים הבאים מצאו את אחת ניצולת המעבד (CPU) הממוצעת במערכת זו:
(נמקו בקצור)

- (1) $Q < S$ קטן מאוד, כמעט אפס.
- (2) $Q = S$
- (3) $S < Q < T$
- (4) $Q > T$
- (5) $Q = \text{"אין סוף"}$

4. (א) הסבירו את הקשר בין עיקרון המקומיות (Locality Principle) והתנהגות תוכניות. כיצד שימוש בעיקרון זה תורם לשיפור ביצועים? הביאו דוגמא לסוג תכנית המתיישב עם עיקרון זה. הביאו דוגמא לסוג תכנית שאינו מתאים לעיקרון המקומיות.

(ב) בזיכרון וירטואלי נתונה מחרחת התייחסות לדפים (page reference string) הבאה, משמאל לימין:

2,3,1,0,3,2,4,3,2,1,0,4

בהנחה שגודל הזיכרון המוקצה לתהליך הוא 4 מסגרות, שרטטו את מפת הזיכרון (איזה מהדפים נמצאים בזיכרון) לאחר כל התייחסות (reference). באיזה מקרים יש תקלת דף, ומה הטפול בה. רשמו את מספר תקלות הדף (page fault) הכולל (בכל אחד מהמקרים):

- (1) LRU
- (2) OPT (האלגוריתם האופטימלי (off-line))
- (3) FIFO

5. (א) תהליכים המתקשרים דרך pipe דורשים לשם כך הקצאת קטנה של שטח בזיכרון. מדוע היא קטנה? מדוע היא נחוצה בכלל?

(ב) מה הפלט של התכנית הבאה? נמקו!

```
#include <stdio.h>

main()
{
    FILE *x;
    char y[10];

    x=popen("echo /tmp", "r");
    fgets(y, BUFSIZ, x);
    y[strlen(y)-1]='\0';
    if (fork()==0)
        execl("/bin/ls", "rm", y, NULL);
}
```