

אוניברסיטת בן-גוריון בנגב, המחלקה למדעי המחשב

מועד א' במערכות הפעלה (27 ביוני 2010)

מרצים: דני הנדלר ואמנון מייזלס; מתרגלים: ליאנה דיזנדרוק, דניאל גורדון, אלון גרובשטיין ואמיר גרשמן

משך המבחן: שלוש שעות
חומר עזר: אסור
פתרי את כל השאלות: סה"כ 100 נקודות.

1. ניהול זיכרון (עבודה מס' 2) (25 נקודות)

- א. בעבודה מס' 2 ההחלטה על החלפת דפים בזכרון נעשתה בעזרת אלגוריתם ה-aging (Not recently used). תארו אלגוריתם זה. (5 נק')
- ב. אלגוריתם זה הינו קירוב ל-LRU. תארו תסריט הממחיש מדוע אלגוריתם ה-aging הינו קירוב בלבד. מדוע לא ניתן להשתמש ב-LRU כאלגוריתם ניהול דפי הזכרון? (4 נק')
- ג. ה-MMU מאפשר להמיר כתובת זכרון וירטואלי לכתובת זכרון פיזי. בעבודה התבקשתם לממש את השיטה של *inverted page table*. ענו על השאלות הבאות:
- מהם המרכיבים המרכזיים של ה-IPT? תארו בקצרה את תהליך החיפוש ב-IPT. (4 נק')
 - תארו את אופן המרת הכתובת מוירטואלית לפיזית במערכת זו. (4 נק')
 - מתי עדיף להשתמש באלגוריתם זה על פני אלגוריתם ה-*page table* הרגיל? נמקו. (4 נק')
 - בעבודה התבקשתם לסנכרן את ה-MMU? מדוע הסנכרון נדרש? (4 נק')

2. מערכות קבצים (35 נקודות)

- א. הסבירי בקצרה מהו ההבדל בין *hard link* ובין *soft link*. (4 נק')
- ב. נתון קובץ *Unix* בשם `/users/studs/Shir/filename` עני בקצרה על השאלות הבאות:
- היכן מאוכסן שם הקובץ `filename`? (3 נק')
 - האם המיקום בו רשום שם הקובץ מאפשר יצירת *hard links*? פרטי מדוע. (3 נק')
 - האם המיקום בו רשום שם הקובץ מאפשר יצירת *soft links*? פרטי מדוע. (3 נק')
 - טענה: "אם המערכת לא הייתה תומכת ב-*hard links* או ב-*soft links* אזי האפשרות הטובה ביותר לשמירת שם הקובץ היא בתוך ה-*i-node*". האם את מסכימה עם הטענה או לא? תני לפחות סיבה אחת טובה התומכת בטענתך. (4 נק')
 - מהו הסיכון כאשר מערכת הקבצים של *Unix* מאפשרת *hard link* ל-*directory*? (6 נק')
 - נתונות זוג מערכות קבצים מבוססות *Unix*, המתקשרות זו עם זו באמצעות פרוטוקול *NFS*.
 - האם ניתן לבצע *soft link* לקובץ הנמצא על מחשב אחד ממחשב אחר המחובר לרשת? הסבירי תשובתך בקצרה ובמדויק. (4 נק')
 - האם ניתן לבצע *hard link* לקובץ הנמצא במחשב אחד ממחשב אחר המחובר לרשת? הסבירי תשובתך בקצרה ובמדויק. (4 נק')
 - הסבירי מה ההבדל בין פעולת *mount* לבין יצירת *link*. (4 נק')

3. סינכרוניזציה - מוניטורים (15 נקודות)

להלן אלגוריתם המממש Producer-consumer buffer בגודל N באמצעות מוניטור.

```
Monitor ProducerConsumer
1. condition full, empty
2. integer count initially 0

3. procedure insert(item: integer)
4. begin
5.     if count=N then wait(full)
6.     insert_item(item)
7.     count=count+1
8.     if count > 0 then signal(empty)
9. end

10.procedure remove: integer
11.begin
12.    if count=0 then wait(empty)
13.    remove=remove_item
14.    count=count-1
15.    if count < N then signal(full)
16.end
end Monitor
```

זכור לכם, מוניטור הוא מטיפוס Hoare אם מתקיימות הדרישות הבאות:
א. כל אימת שמתקבל סיגנל על ידי חוט (thread) t, מובטח כי t הוא החוט הבא שיכנס למוניטור.
ב. פעולת signal היא האחרונה וחוט המבצע אותה יוצא מיד אחר כך מן המוניטור.

הוכיחו בקצרה את הטענה הבאה, או תנו תסריט מדוייק הסותר אותה:

המימוש שלעיל נכון גם אם המוניטור אינו מטיפוס Hoare.

4. סינכרוניזציה – בעיית המניעה ההדדית (mutual exclusion) (25 נקודות)

א. להלן אלגוריתם למניעה הדדית עבור שני תהליכים תהליך 0 ותהליך 1.

```
shared int b[0], b[1] both initially 0
Code for process  $i \in \{0,1\}$ 
1. b[i] := 1;
2. while b[1-i] = 1 do
3.   b[i] := 1-i;
4.   await b[0] = 1-i;
5.   b[i] := 1;
6. od;
7. critical section
8. b[i] := 0;
```

i. האם הקוד לעיל מקיים mutual exclusion? הוכיחו במדויק או הפריכו ע"י מתן תסריט מדויק (10 נק').

ii. האם הקוד לעיל מקיים deadlock-freedom? הוכיחו במדויק או הפריכו ע"י מתן תסריט מדויק (5 נק').

iii. האם הקוד לעיל מקיים starvation-freedom? הוכיחו במדויק או הפריכו ע"י מתן תסריט מדויק (5 נק').

ב. בכיתה הגדרנו תכונה של אלגוריתמי mutual exclusion הקרויה First-in-first-out (FIFO) באופן הבא:

"אם תהליך p ממתין ותהליך q טרם סיים את ה-doorway אזי q אינו יכול להיכנס לקטע הקריטי לפני p ."

הוכיחו או הפריכו בקצרה אך במדויק את הטענה הבאה:

"אלגוריתם למניעה הדדית המקיים מניעה הדדית ו-FIFO מקיים גם starvation freedom". (5 נק')

בהצלחה!

