



אוניברסיטת בן-גוריון בנגב

מדור בחינות

תאריך הבחינה: 18.7.06  
 שם המורה: בראך מיצ'לס + מר גילן  
 מבחן ב: מזיכרון הבצ'ה  
 מסי הקורס: 202-1-3031  
 מיועד לתלמידי: מגזר-הממשל  
 שנה: ג' סמ': ב' מועד: ב'  
 משך הבחינה: 3 שאלה  
 חומר עזר: !/ok

מסי מבחן: \_\_\_\_\_

ענה/י על כל חמש השאלות. כתוב/י בכתב ברור ונקי והסבר/י את תשובותיך. הניקוד מסומן ליד כל שאלה, סכום כל הנקודות הוא 100.

20 נק') 1. נתונה התוכנית הבאה לפתרון בעיית הפילוסופים האוכלים" (עבור  $N=5$ ).

```
void philospher(int i)
{
    while(TRUE)
    {
        think();
        pick_stick(i);
        pick_stick((i+1) % 5);
        eat();
        put_stick(i);
        put_stick((i+1) % 5);
    }
}
```

עני על השאלות הבאות:

- א. האם הפתרון הוא Deadlock-free? אם כן, הוכיחי, אחרת הראי תסריט שייצור Deadlock. (5 נק')
- ב. הקוד בסעיף א' הוא Uniform כלומר זהו עבור כל אחד מהפילוסופים. אם היינו יכולים לכתוב קוד שונה לכל פילוסוף, האם פיתרון שבו יש לכל מקל semaphore בינארי המייצג אותו היה אפשרי? הוכיחי שלא, או רישמי את הקוד עבור הפילוסופים (5 נק').
- ג. עני על שני הסעיפים הבאים:
  1. מהם השינויים הנדרשים על מנת לפתור את בעיית הפילוסופים בעזרת אלגוריתם הבנקאי? הסבירי במדויק (5 נק').
  2. מדוע לדעתך לא הוצע פיתרון זה עד היום? (5 נק')

20 נק') 2. שאלה זו מתיחסת לתרגיל תכנותי מס' 1

- א. נניח כי מידע אודות קודקוד מספר 5 נמצא בטבלאות (RT) של כל הקודקודים, אך לפתע הקודקוד הזה נופל. תארי כיצד (בסופו של דבר) יתעדכנו הטבלאות של כולם כך שבאף אחת מהן לא יופיע קודקוד מספר 5 (5 נק')
- ב. נניח שלא היה קיים mailer וכל קודקוד פשוט היה מכניס את ההודעות שלו לתיבת הדואר של הקודקוד אליו מיועדת ההודעה במקום להכניס אותה לתיבת הדואר של ה-mailer. האם שינוי זה בעייתי? נמקי (5 נק')

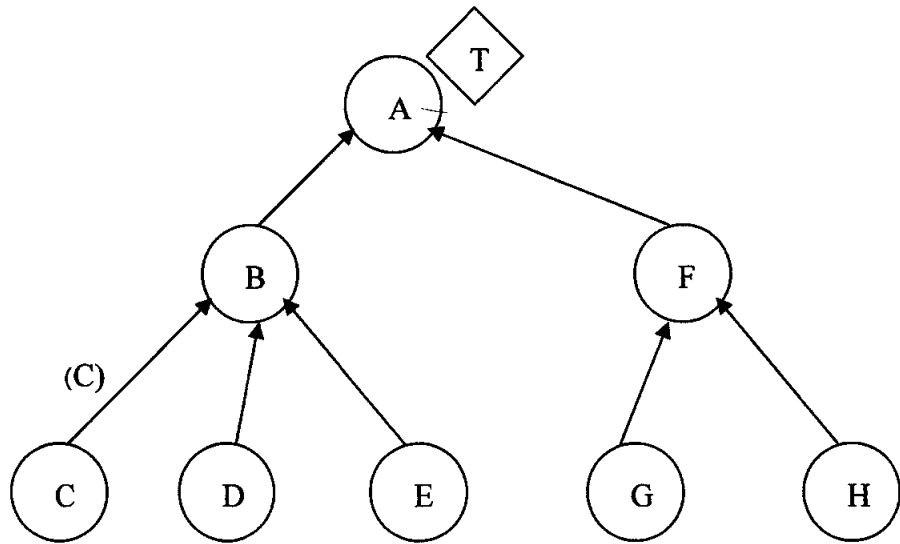
- ג. בתרגיל נתבקשת לתמוך בפקודה `crashNode NodeId` אשר אמורה לגרום לסימולטור "לרסק/לכבות" קודקוד מסוים, אלא אם כן המערכת עלולה להיכנס למצב של `deadlock` בעקבות ביצוע הפקודה. תארי מצב כני"ל, שבו אסור לבצע את הפקודה מחשש ל-`deadlock`. (3 נק')
- ד. הציעי שינוי לסימולטור כך שבנוסף למה שכל קודקוד "מגלה" עכשיו (קרי – המרחקים הקצרים ביותר לכל קודקוד), הוא גם יגלה את מבנה הגרף כולו (ז"א יידע את השמות – `id` – של כל הקודקודים בגרף ויידע בדינא איזה קשתות קיימות ביניהם). כיצד ניתן לשנות/להוסיף לאלגוריתם ע"מ שכל קודקוד יגלה גם זאת (7 נק')

### 20 נק') 3. ניהול זכרון

- א. הוכח כי LRU הוא אלגוריתם מחסנית (3 נק').
- ב. כיצד ניתן להוכיח כי FIFO אינו אלגוריתם מחסנית (2 נק').
- ג. עבור ה-`ref-string` הבאה, ושימוש באלגוריתם LRU, מהי כמות הזיכרון המינימלית הנדרשת על מנת שלא יהיו יותר מ-10% Page-Faults מעבר למספר ה-PF האופטימלי (7 נק').  
7, 2, 5, 3, 4, 3, 4, 3, 7, 2, 5, 4, 3, 4, 3, 1, 2, 3, 4, 5, 6, 7
- ד. מדוע נדרשת טבלת העוגן במנגנון ה-HASHING של ה-Inverted-Tables (8 נק').

### 20 נק') 4. סנכרון מבוזר:

- נתונה רשת בת 8 סוכנים המסודרת בעץ שבציור ואשר מריצה את האלגוריתם של Raymond להעברת אסימון, כדי להבטיח `mutual exclusion`. במצב ההתחלתי האסימון נמצא אצל הסוכן A שבשורש העץ ואין בקשות להכנס לקטע הקוד הקריטי. כיוון הקשת היוצא מכל סוכן בעץ שבציור מתאר את המצביע `current_direction` של האלגוריתם. האסימון נמצא אצל הסוכן A ומסומן כמעוין. כדי לתת תאור נח לשלבי הריצה של האלגוריתם, נגדיר צעדים. אצל כל סוכן, צעד כולל הפעלה של איזשהי פרוצדורה. למשל, משלוח של הודעה או קבלה של הודעה וכמובן החישוב הנדרש הכולל הכנסה או הוצאה מהמחסנית. משלוח של הודעה מתואר בציור על ידי התוכן שלה בסוגריים. בציור ישנה הודעה (שאיננה חלק מהמצב ההתחלתי המתואר) הנשלחת על ידי סוכן C אל סוכן B ותוכנה הוא C. המשלוח הוא על פי כיוון הקשת היוצאת מסוכן C בציור.
- במצב ההתחלתי המתואר ביקשו הסוכנים C, D, G, F את קטע הקוד הקריטי. תארי את צעדי הריצה הבאים של האלגוריתם, לפי הגדרת הצעד שלמעלה. בכל צעד בתאור שלך, כל אחד מן הסוכנים מבצע צעד יחיד (פעולה יחידה). בכל צעד נדרש לתאר את ההודעות הנשלחות, את תוכן המחסניות, את כיוון המצביעים, ואת מיקום האסימון. הניחי שכל סוכן שביקש את קטע הקוד הקריטי מחזיר אותו בצעד הבא. כמו כן הניחי כי אם מגיעות שתי הודעות לאותו סוכן באותו צעד, הוא מטפל בהן בסדר הלכסיקוגרפי של השולחים. **תני את תאור הצעדים עד שהסוכן השלישי נכנס לקטע הקוד הקריטי.**



20 נק' 5. נתונה התוכנית הבאה:

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>

//#define _REENTRANT

int main (char **argv, int argc)
{
    char buf[100];
    int fd = open("dugma1.txt",O_WRONLY,0666);
    int fd2 = dup(fd);    1111111111
    write(fd,"I was here first",16);
    write(fd2,"I was here second",17);
}
}

```

א. בהנחה שבתחילת התוכנית הקובץ `dugma1.txt` הוא ריק. מה תוכנו של הקובץ לאחר סיום ריצת התוכנית? הסבירו (5 נק').

ב. החליפי את השורה המסומנת באחדות על מנת לקבל תוצאה שונה והסבירו את השוני, בהתייחס למימוש מערכת הקבצים של UNIX (5 נק').

ג. מה היה משתנה בכל אחד משני הסעיפים הקודמים אם לאחר השורה המסומנת באחדות היינו מוסיפים את השורה:

```
lockf(fd, F_LOCK, 16);
```

הסבירי (5 נק').

הערה: הפרמטר השני של הפקודה lockf יכול לקבל ארבעה ערכים:

F\_ULOCK – מסיר נעילה.

F\_TEST – בודק האם יש נעילה על הקטע.

F\_LOCK – נועל קטע אם הוא פנוי. אם לא, ישן עד שמתפנה.

F\_TLOCK – נועל קטע אם הוא פנוי. אם לא, מחזיר -1.

ד. מה היה משתנה בכל אחד משני הסעיפים א, ו ב אם בנוסף לשורה שנוספה בסעיף ג' נוסיף בין שתי שורות ה - write את השורה:

```
lockf(fd, F_TLOCK, 17);
```

הסבירי (5 נק').

**בהצלחה!**

