



8/7/09

מרצה: ד"ר עופר שטריכמן  
משך המבחן: שעתיים וחצי.

יש לכתוב את התשובות על דפים אלה. מותר (רצוי!) בעפרון. חומר מותר: 2 דפים כתובים בכתב ידכם.

מבוא לטכנולוגיות מחשב – מבחן מסכם

1. (35 נק') להלן תוכנית ובה לכל איבר 4 מספרי קלט ו 4 מספרי פלט. התוכנית רצה על מחשב עם זיכרון מטמון המיועד ל data בגודל 512 בתים, ובלוק בגודל 32 בתים. זיכרון המטמון עובד בשיטה של הוצאת ה Least Recently Used. מבנה הנתונים הינו:

```
struct MyIO {
    int in[4];
    int out[4];
};
struct MyIO IO_array[16][16];
register int i, j, k;
```

הניחו ש:

sizeof(int) = 4

-- זיכרון המטמון בתחילת הריצה ריק.

-- הגישות היחידות לזיכרון הם ל IO\_array. משתנים i, j, k, temp0...3 מאוחסנים ברגיסטרים.  
-- זמן גישה לזיכרון: במקרה של 100 cache miss מחזורי שרון, במקרה של 20 cache hit מחזורי שרון.

א. עבור התוכנית הבאה:

```
for (j=0; j<16; j++){
    for (i=0; i<16; i++) {
        temp0=IO_array[i][j].in[0];
        temp1=IO_array[i][j].in[1];
        temp2=IO_array[i][j].in[2];
        temp3=IO_array[i][j].in[3];
        IO_array[i][j].out[0]=temp0++;
        IO_array[i][j].out[1]=temp1++;
        IO_array[i][j].out[2]=temp2++;
        IO_array[i][j].out[3]=temp3++;
    }
}
for (i=0; i<16; i++){
    for (j=0; j<16; j++) {
        for (k=0; k<=3; k++) {
            IO_array[i][j].out[k] +=2;
        }
    }
}
```

קטע קוד ראשון

קטע קוד שני

(7) מספר מחזורי שרון המיועדים לגישות לזיכרון בקטע הקוד הראשון הוא (הצג חישוב):

(7) מספר מחזורי שרון המיועדים לגישות לזיכרון בקטע הקוד השני הוא (הצג חישוב):

(7) ה miss-rate הכולל לפעולות כתיבה ל IO\_array (כלומר, אחוז ה cache-misses מסך הפעולות, עבור קטעי הקוד הראשון והשני) הוא (הצג חישוב): \_\_\_\_\_

ב. התוכנית נכתבה בשנית עם מבנה נתונים שונה:

```
struct MyIN {
    int in[4];
};
struct MyOUT {
    int out[4];
};

struct MyIN IN_array[16][16];
struct MyOUT OUT_array[16][16];
```

ובאותן הנחות:

```
for (j=0; j<16; j++){
    for (i=0; i<16; i++) {
        temp0=IN_array[i][j].in[0];
        temp1=IN_array[i][j].in[1];
        temp2=IN_array[i][j].in[2];
        temp3=IN_array[i][j].in[3];

        OUT_array[i][j].out[0]=temp0++;
        OUT_array[i][j].out[1]=temp1++;
        OUT_array[i][j].out[2]=temp2++;
        OUT_array[i][j].out[3]=temp3++;
    }
}

for (i=0; i<16; i++){
    for (j=0; j<16; j++) {
        for (k=0; k<3; k++) {
            OUT_array[i][j].out[k]+=2;
        }
    }
}
```

} קטע קוד ראשון

} קטע קוד שני

(7) שינוי זה ישפר / לא ישפיע / יגרע מביצועי קטע קוד 1. הסבר: \_\_\_\_\_

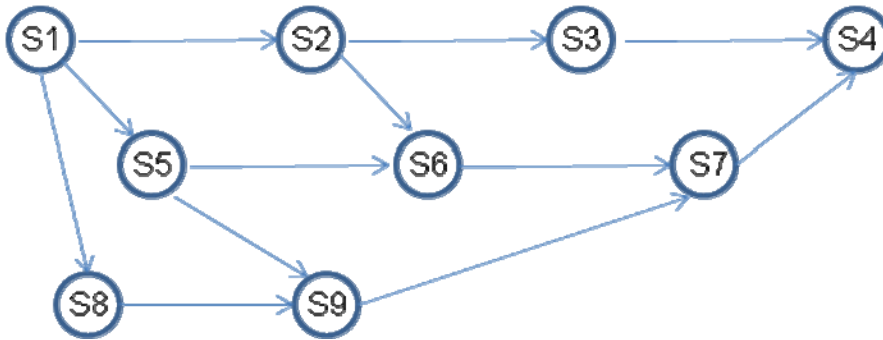
(7) שינוי זה ישפר / לא ישפיע / יגרע מביצועי קטע קוד 2. הסבר: \_\_\_\_\_



4. (25)

גרף אילוצים (constraints graph) הינו גרף מכוון וחסר מעגלים המגדיר סדר חלקי של ביצוע שורות קוד. הצמתים של גרף אילוצים מיצגים שורות קוד. קשת מכוונת מצומת  $v$  אל צומת  $u$  מביעה את האילוץ כי שורת הקוד המיוצגת ע"י  $v$  צריכה להתבצע לפני שורת הקוד המיוצגת ע"י  $u$ .

כך, למשל, בגרף האילוצים שלהלן, שורת הקוד המיוצגת ע"י S1 צריכה להתבצע לפני שורות הקוד המיוצגות ע"י S2, S5 ו-S8, ואילו שורת הקוד המיוצגת ע"י S6 צריכה להתבצע אחרי שורות הקוד המיוצגת ע"י S2 ו-S5.



להלן קטעי קוד המבוצעים ע"י שני תהליכים A ו-B.

<u>Process A</u>	<u>Process B</u>
S1	S2
S5	S3
S8	S6
S9	S4
S7	

א. (15 נקודות) הגדירו שני semaphores, והוסיפו פעולות wait ו-signal על semaphores אלו במקומות המתאימים בקוד של תהליכים A ו-B (הוסיפו פקודות על דף הבחינה, בין השורות לעיל), כך שבכל ריצה של שני התהליכים סדר ביצוע שורות הקוד יהיה בהתאם לגרף האילוצים הנתון. פתרון המשתמש ביותר משני semaphores יזכה בניקוד חלקי.

ב. (10 נקודות) תנו נימוק קצר ומדויק לכך שלא ניתן לקיים את האילוצים המיוצגים ע"י הגרף לעיל ע"י semaphore יחיד.

---



---



---



---