



תאריך הבחינה: 21.6.04
 שם המרצה: פרופ' אהוד גודס
 ומר רועי זיוון
 מבחן ב: מערכות הפעלה
 מס' קורס: 202-1-3031
 שנה: תשס"ד סמ': ב' מועד: א'
 משך הבחינה: 3 שעות
 חומר עזר: אסור

פתור את כל השאלות, סה"כ 100 נקודות.

1. (25 נק')
 נתון פיתרון בעיית המינהרה.
 א. שנה/י את הפיתרון כך שתינתן עדיפות למכוניות המגיעות מצד "אפס". מכוניות מצד "אחד" נכנסות רק אם אין מכוניות ממתניות בצד "אפס".
 ב. שנה/י את הפיתרון (הנתון) כך שלא תיתכן הרעבה. במידה וממתניות מכוניות משני הכיוונים הן יכנסו לסירוגין. אם רק מכוניות מצד אחד מעוניינות להיכנס, הן יכולות להיכנס ללא הגבלה.
הערה: בשני הסעיפים ניתן לשנות את הפיתרון עבור אחד הכיוונים! כמו כן יש להשתמש במונים מסוג Integer וסמפורים בינאריים בלבד.

One-way tunnel - solution

```

int          count[2];
semaphore   mutex = 1, busy = 1;
semaphore   waiting[2] = {1,1};

void arrive(int direction) {
    down(&waiting[direction]);
    down(&mutex);
    count[direction] += 1;
    if(count[direction] == 1)
        up(&mutex);
    down(&busy);
    else up(&mutex);
    up(&waiting[direction]);
}

void leave(int direction){
    down(&mutex);
    count[direction] -= 1;
    if(count[direction] == 0)
        up(&busy);
    up(&mutex);
}

```

2. (10 נק')
נתון המימוש של בעיית ספק לקוח באמצעות מוניטורים הראה סצנריו שיגרום לכך שגודל הבאפר האפקטיבי יהיה 1 במקום n.

Bounded-Buffer with Monitors

```
monitor ProducerConsumer
condition full, empty;
integer count;
procedure enter;
begin if count = N then wait(full);
      enter_item;
      count := count + 1;
      if count = 1 then signal(empty); end
procedure remove;
begin if count = 0 then wait(empty);
      remove_item;
      count := count - 1;
      if count = N - 1 then signal(full); end
count := 0;
end monitor
```

תזכורת: במוניטור הנ"ל, תהליך מוער מוכנס לתוך המוניטור ולכן הפקודה Signal חייבת לסיים פרוצדורה (אינה יכולה להופיע באמצע).

3. (15 נק')
מערכת קבצים:
- מדוע בנוסף למנגנון אבטחת נתונים, מספק Unix למשתמשיו מנגנון נעילת קבצים כגון `lock f()`? (עד 5 שורות).
 - מדוע במערכת קבצים מבוזרת NFS, מנגנון נעילה כזה אינו אפשרי? (עד 5 שורות).
 - הצע פתרון לבעיה בסעיף ב', בלי לשנות את מימשק NFS (שלוש הפונקציות Write, read, lookup).
 - אילו סכנות יש בפיתרון שהצעת (עד 3 שורות).

4. (20 נק') 20

שאלה זו מתייחסת לתרגיל תכנותי מספר 2. (אין קשר בין הסעיפים השונים).

- א. נדרשתם בתרגיל לתמוך בשלושה תזמונים שונים של ה-scheduler (round robin, longest queue, application specified). תארי בקצרה את אלגוריתם התזמון עבור Application specified בו השתמשת.
- ב. ענה/י נכון/לא נכון על השאלות הבאות ונמקי בקצרה את תשובתך:
1) אם גודל תיבת הדואר של ה-Mailer חסום ואילו גודל תיבות הדואר של הקודקודים אינו חסום, עלול להיווצר מצב של-Deadlock?
2) הורדת צלע מהגרף תגרום ליותר הודעות להשלח ברשת (הגרף נותר כשיר גם לאחר ההורדה של הצלע, וכל ניתוב של הודעה ע"י קודקוד "ביניים" נחשב לשליחת הודעה נוספת).
ג. נניח כי המערכת שונתה כך שעבור כל שליחת הודעה בין כל שני קודקודים סמוכים ברשת, הקודקוד השולח ימתין לאישור מה-Thread של ה-Mailer לכך שההודעה נשלפה על ידיו מתיבת הדואר שלו. האם עלול להיווצר Deadlock? הוכח אם לא או הבא דוגמא אם כן.
הערה: כל תהליך שולח מחכה על סמפור נפרד לאישור מה-Mailer.

5. (25 נק') 25

נתונה סדרת דרישות דפים מן הזיכרון (Reference-string) הבאה, משמאל לימין:

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

- א. אם משתמשים באלגוריתם החלפת דפים LRU, כתוב/י את ה-Distance string עבור הסדרה הנתונה. חשבי/י מתוך ה-Distance string כמה Page-faults יהיו עבור גודל זיכרון פיזי בן 4 דפים, האם כדאי להגדיל את הזיכרון הפיזי ל-5 דפים בדוגמא זו?
- ב. הנח כי מממשים את אלגוריתם ה-LRU באמצעות Shift Registers (aging) של 4 ביטים. רגיסטרים אלו מתעדכנים כל 4 יחידות זמן. הראה את מצב הרגיסטרים עבור כ"א מהדפים בזמנים 1 עד 8. האם אלגוריתם זה יממש LRU במדוייק, במקרה זה? תמיד?
- ג. תאר מימוש של הפקודות fork ו-execvp (כל אחת בניפרד) במערכת Unix, המנסה לחסוך מקום עד כמה שאפשר על הדיסק (ב-Paging file).

6. (5 נק') .

הפקודה CHOWN ב-Uix יכולה לשנות את ה- Owner של קובץ. נניח שפקודה זו יכולה להתבצע ע"י כל משתמש. תאר סצנריו שבאמצעותו משתמש רגיל יכול למחוק קבצי מערכת מוגנים.

בהצלחה !