

אוניברסיטת בן-גוריון בנגב, המחלקה למדעי המחשב
מועד א' במערכות הפעלה (11 באוגוסט 2008)

מרצה: דר' דני הנדלר; מתרגלים: דניאל גורדון, אלון גרובשטיין, ליאנה דיזנדרוק; בודקים: עדי סוויסה, שי ליברמן

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

1. תזמון תהליכים (25 נקודות)

הטבלה הבאה מתארת קבוצת תהליכים במערכת.

תהליך	Burst time	זמן הגעה
P1	23	0
P2	8	3
P3	17	18
P4	12	20

- א. הניחו כי ה-scheduler של המערכת משתמש באלגוריתם Round robin עם time quantum של 10 יחידות זמן. הניחו גם כי תהליך חדש המגיע למערכת נכנס למקום הראשון בתור של אלגוריתם ה-Round robin. שרטטו Gantt table מתאימה וחשבו את ה-turnaround time וה-waiting time הממוצעים. פרטו את חישוביכם.
- ב. הניחו כעת כי ה-scheduler משתמש באלגוריתם Preemptive Shortest Job First (הידוע גם בכינויו Shortest remaining time first). שרטטו Gantt table מתאימה וחשבו את ה-turnaround time הממוצע. פרטו את חישוביכם.
- ג. הזכרנו בקורס מספר מקרים בהם schedulers מעדכנים באופן דינמי את ה-priorities של תהליכים. תנו שתי דוגמאות למקרים בהם ה-scheduler עושה זאת והסבירו במדויק עבור כל אחת מן הדוגמאות מהו ההגיון שמאחורי עדכון דינמי זה. (אין צורך לציין איזו מערכת הפעלה מפעילה מנגנון זה.)

2. ניהול זיכרון (25 נקודות)

- א. נתונה מערכת ובה מרחב הזיכרון הוירטואלי הינו של 48-ביטים, מרחב הזיכרון הפיזי הינו של 32-ביטים, וגודלו של כל דף הינו 4K. בהנחה כי טבלת הדפים אינה multi-level כמה דפים היא מכילה? כמה page frames יש בזיכרון הפיזי?
- ב. מהי טבלת דפים הפוכה (inverted page table)? ציינו יתרון אחד וחסרון אחד של טבלת דפים הפוכה ביחס לטבלת דפים רגילה (שאינה multi-level).
- ג. הסבירו בקצרה מהו TLB ומהו תפקידו במערכת ניהול זיכרון.
- ד. נתונה מערכת ובה ארבעה page frames. הטבלה הבאה מתארת את מצב הזיכרון הפיזי בזמן נתון.

דף (page frame)	זמן טעינה	זמן גישה אחרונה	R bit	Modified bit
1	126	280	1	0
2	230	265	0	1
3	140	270	0	0
4	110	285	1	1

ציינו אילו מבין הדפים שלעיל יהיה הראשון שיוחלף ע"י כל אחד מן האלגוריתמים הבאים. נמקו תשובתכם בקצרה. הניחו כי הסריקה מתחילה בדף מספר 1.

- (a) NRU
 - (b) FIFO
 - (c) LRU
 - (d) Second-chance FIFO
- n

3. מערכת הקבצים (25 נקודות)

א. מערכות הפעלה שונות משתמשות במבני נתונים שונים על מנת למפות את הבלוקים הפנויים בדיסק. מערכת קבצים הינה אמינה יותר אם במקרה של אבדן אינפורמציה זו היא יכולה לשחזר אותה על סמך מבני נתונים אחרים.

בתשובתכם הניחו כי מבנה הנתונים הממפה את הבלוקים הפנויים בדיסק אבד בשלמותו.

- a. האם במערכת Unix ניתן לשחזר אינפורמציה זו? אם כן הסבירו כיצד, אם לא נמקו מדוע.
- b. האם במערכת FAT ניתן לשחזר אינפורמציה זו? אם כן הסבירו כיצד, אם לא נמקו מדוע.

ב. כזכור לכם, רשומת i-node כוללת שדה בשם count, המונה את מספר ה-hard links המצביעים לקובץ. נניח כי נפילת מתח גורמת לתקלה בה הערך בשדה ה-count שונה ממספר ה-hard links המצביעים לקובץ בפועל.

- a. אילו בעיות עלולות להיגרם אם ערכו של שדה ה-count **קטן** ממספר ההצבעות בפועל? איך ניתן לפתור תקלה כזו?
- b. אילו בעיות עלולות להיגרם אם ערכו של שדה ה-count **גדול** ממספר ההצבעות בפועל? איך ניתן לפתור תקלה כזו?

ג. אחד הפרמטרים המשפיעים על ביצועי מערכת הקבצים הינו גודל הבלוק אשר בו היא משתמשת. פרטו את היתרונות והחסרונות של השימוש בבלוקים גדולים לעומת בלוקים קטנים. בפרט, התייחסו לשאלה איזה גודל בלוק עדיף בגישות סדרתיות (sequential) לקבצים ואיזה עדיף בגישות random access.

4. סינכרוניזציה (25 נקודות)

א. בסעיף זה עליכם לממש counting semaphore המאותחל לאפס תוך שימוש במוניטור רגיל (כלומר, לא מוניטור של Java). ה-counting semaphore אותו תממשו אינו חייב להיות הוגן. עליכם להשלים את מימוש הפרוצדורות up ו-down. **כיתבו את מימוש פרוצדורותאלו במחברת** אין להגדיר משתנים נוספים. על הפתרון לעבוד ללא קשר אם לאחר שתהליך ביצע פעולת signal על ה-condition variable הוא ממשיך לרוץ או לא וללא קשר אם התורים של המוניטור הוגנים או לא.

Monitor MyCountingSemaphore

```
condition variable mutex
int counter=0

procedure up() {
    .
    . (השלימו קוד זה במחברת)
    .
}
```

```

procedure down() {
    .
    . (השלימו קוד זה במחברת)
    .
}

```

ב. בסעיף זה עליכם לממש מוניטור רגיל (כלומר, לא מוניטור של Java) תוך שימוש במשתני `integer` ובסמאפורים. לשם פשטות, המוניטור עליכם לממש מכיל פרוצדורה יחידה `my-proc` וכן `condition variable` יחיד `X`. להלן תבנית קוד המממשת מוניטור זה. עליכם להשלים את הקוד של הפרוצדורה `monitor-proc` העוטפת את הפרוצדורה `my-proc`. כמו כן, עליכם להשלים את הקוד של הפרוצדורות `X-wait` ו-`X-signal` המממשות פעולות `wait` ו-`signal` על `condition variable X`. כיתבו את מימוש פרוצדורות אלו במחברת אין להוסיף מישתנים נוספים. הניחו כי הפרוצדורות **X-signal** ו-**X-wait** נקראות רק מתוך הפרוצדורה **my-proc**.

```

binary semaphore mutex=1
counting semaphore semX=0
int countX=0

```

```

procedure monitor-proc(){
    _____ (השלימו שורה זו במחברת)

my-proc()
    _____ (השלימו שורה זו במחברת)

}

```

```

procedure X-wait()
    .
    . (השלימו קוד זה במחברת)
    .
}

```

```

procedure X-signal()
    .
    . (השלימו קוד זה במחברת)
    .
}

```

בהצלחה !