

אוניברסיטת בן-גוריון בנגב, המחלקה למדעי המחשב

מועד ב' במערכות הפעלה (2 בספטמבר 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)

- א. פרופסור מוטי פשקובסקי, גסטרונום וינין בהכשרתו, החליט להתמחות גם במערכות הפעלה. במסגרת המתחותו החדשה, הוא הכניס שינוי במערכת Unix. כזכור לכם, במערכת Unix קיימת פונקציית `wait()` אשר באמצעותה יכול תהליך להמתין עד אשר תהליך בן (`child process`) שלו מסיים את ריצתו. במקום המימוש הקיים של `wait`, מימש פרופסור פשקובסקי פרוצדורה בשם `child_is_alive()`. פרוצדורה זו מחזירה מייד `True` אם התהליך הבן עדיין חי ומחזירה מייד `False` אם התהליך הבן הסתיים.
- ממשו את הפרוצדורה `wait()` תוך שימוש בפרוצדורה `child_is_alive`.
ii. השוו את יעילות המימוש החדש של `wait` למימוש הרגיל של Unix. הצדיקו תשובתכם.
- ב. פרופסור פשקובסקי החליט להכניס שינוי ב-Scheduler של Unix. במקום המימוש הקיים והמורכב של `multi-level priority queues`, הוא החליט לממש אלגוריתם `Non-preemptive SJF`. בהנחה כי עבור כל תהליך ניתן לדעת תמיד את משך ה-CPU burst שלו וכי `CPU bursts` אלו תמיד קצרים מן ה-CPU quantum של Unix, האם המימוש החדש ישפר או ירע את ה-Turnaround time הממוצע? איך ישפיע המימוש החדש על זמן ההמתנה במיקרה הגרוע (`worst-case waiting time`)? הצדיקו תשובתכם.
- ג. פרופסור פשקובסקי עשה גם נסיון שלישי לשפר את מערכת Unix. במערכות Unix רגילות, משך ה-`time quantum` הניתן לתהליכים הינו כ-10 אלפיות השניה. פרופסור פשקובסקי מימש Scheduler חדש המשתמש באלגוריתם `Preemptive round-robin` הפועל על תור יחיד, עם `time quantum` קצר של מיליונית השניה. מה צפויה להיות ההשפעה של המימוש החדש על ביצועי המערכת? הצדיקו תשובתכם.

2. ניהול זיכרון (25 נקודות)

א. נתונה מערכת בה הזיכרון הפיזי מכיל ארבעה page frames. החל ממצב התחלתי בו page frames אלו ריקים, רץ תהליך יחיד המבצע את סידרת הגישות הבאה לדפים בזיכרון הוירטואלי שלו (סדר הגישות הינו משמאל לימין, כלומר הגישה הראשונה היא לדף 1 והאחרונה לדף 4):

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

עבור כל אחד מן האלגוריתמים הבאים להחלפת דפים, ציינו עבור כל גישה האם היא גורמת ל- page fault ואילו דפים נמצאים בזיכרון לאחר הגישה. עשו זאת על ידי מילוי טבלה כלהלן **במחברת**.

גישה	Page fault?	דפים בזיכרון לאחר הגישה
1	כן	1
2	כן	1, 2

-
-
-

- i. האלגוריתם האופטימלי (optimal page replacement algorithm). **כאשר ישנם מספר דפים אפשריים להחלפה, הניחו כי מוחלף הדף שמספרו מינימלי?**
- ii. אלגוריתם LRU
- iii. FIFO
- iv. FIFO second chance. הניחו כי ה-reference bit מורם כאשר דף מובא לזיכרון וכי האלגוריתם מתחיל להתבצע מן ה-page frame הראשון.

ב. מערכות הפעלה ישנות השתמשו במנגנון של swapping. הסבירו בקצרה את ההבדלים בין swapping לבין paging. ציינו מהם היתרונות והחסרונות של כל אחת משתי השיטות.

3. מערכת הקבצים (25 נקודות)

- א. למערכות הפעלה שונות ישנן דרכים שונות להקצאת זיכרון עבור קבצים בדיסק. להלן שלוש מן השיטות להקצאת זיכרון עבור קבצים בדיסק אותן תיארו בכיתה.
 - a. הקצאה של זיכרון רציף בדיסק (contiguous allocation).
 - b. רשימה מקושרת של בלוקים (עם מצביע בתוך הבלוק).
 - c. רשימה מקושרת הנשמרת בטבלת FAT בזיכרון.

ג. תאר את היתרונות והחסרונות של כל אחד משיטות אלו. אילו מן השיטות עדיפה עבור גישה סידרתית לקובץ? אילו מן השיטות עדיפה עבור גישות random access?

ב. כזכור לכם, במערכת Unix קובץ מתואר ע"י רשומת i-node, ואילו ב-NTFS קובץ מתואר ע"י רשומה של MFT (Faster File Table). ציינו לפחות שלושה הבדלים בין שתי רשומות אלו.

ג. ישנן מערכות קבצים, כגון NTFS, התומכות בדחיסה של קבצים באופן שקוף לאפליקציה: הקובץ נדחס לפני שהוא נכתב לדיסק ונפתח (decompressed) אחרי היקראו מן הדיסק. הסבירו את ההשלכות (הן החיוביות והן השליליות) של דחיסת קבצים על ביצועי המערכת. מהי ההשפעה שיש לדחיסת קבצים על גישות random access לקובץ?

4. סינכרוניזציה (25 נקודות)

להלן אלגוריתם למניעה הדדית עבור n תהליכים.

integer k (initially 1)
boolean $b[1..n]$ (initially true), $c[1..n]$ (initially true)

Code for process i

```
1   $b[i] := false$ 
2  L: while  $k \neq i$  do
3       $c[i] := true$ 
4      if  $b[k]$  then  $k := i$ 
5  od
6   $c[i] := false$ 
7  for  $j := 1$  to  $n$  do
8      if  $j \neq i$  and  $\neg c[j]$  then goto L
9  od
10 Critical Section
11  $c[i] := true$ 
12  $b[i] := true$ 
```

- א. האם האלגוריתם מקיים מניעה הדדית? הצדיקו תשובתכם.
ב. האם האלגוריתם מקיים חופש מקיפאון (deadlock freedom)? הצדיקו תשובתכם.
ג. האם האלגוריתם מקיים חופש מהרעבה (starvation freedom)? הצדיקו תשובתכם.

בהצלחה!