מס' מחברת_____

מס' ת.ז._____

האוניברסיטה העברית בירושלים
ביה"ס להנדסה ומדעי המחשב

מבחן במערכות הפעלה

קורס מס' 67808

מועד א' תשס"ט
המורה: פרופ' דרור פייטלסון

תאריך: 13.7.09
זמן: 2.5 שעות

במבחן שני חלקים. בחלק הראשון יש לענות על 11 מתוך 12 שאלות. משקל כל שאלה 5 נקודות. בחלק השני יש לענות על 3 מתוך 4 שאלות, שמשקל כל אחת מהן 15 נקודות. מומלץ להשתמש במחברת לטיוטה. יש לכתוב את התשובות הסופיות בעט בטופס המבחן *בצורה נקייה ומסודרת.*

בהצלחה!

חלק א' (55 נקודות)

ענה על 11 מתוך 12 השאלות הבאות. בכל שאלה יש להקיף בעיגול את התשובה הנכונה. נא לסמן באופן ברור את השאלה שאין לבדוק על ידי מתיחת קו אלכסוני על כל השאלה.

1. first fit ו-best fit הן שיטות להקצאת קטעי זכרון רציפים בגדלים שונים.
   א) זמן הריצה של best fit תמיד ארוך יותר משל first fit
   ב) best fit מגן טוב יותר נגד פרגמנטציה פנימית
   ג) בהינתן מצב התחלתי מסוים ורצף בקשות להקצאה, אם first fit מסוגל לספק אותן אזי בהכרח גם best fit מסוגל לכך דוגמה סותרת: יש מקטעים פנויים בגדלים 30 ו-20, ורצף בקשות ל-10, 20, ועוד 20.
   ד) העדיפות של שיטה אחת על פני חברתה מבחינת היכולת לבצע הקצאות תלויה במצב ההתחלתי וברצף הבקשות

2. צורה אפשרית לנהל זכרון של תהליך היא להשתמש בטבלת סגמנטים, כאשר לכל סגמנט יש טבלת דפים נפרדת והדפים מועברים בין הזכרון לדיסק לפי הצורך. במערכת כזו, איזו מהבעיות הבאות לא יכולה לקרות:
   א) exception כי הסגמנט המבוקש אינו מסומן בתור valid בטבלת הסגמנטים
   ב) exception כתוצאה מחריגה מגודל הסגמנט
   ג) page fault כי הדף המבוקש אינו מסומן בתור valid בטבלת הדפים
   ד) exception כתוצאה מחריגה מגודל הדף

3. מבין האלגוריתמים הבאים להחלפת דפים במערכות זיכרון וירטואלי, איזה צפוי להקטין את מספר ה-page faults למינימום? הנח כי כל האלגוריתמים אכן ניתנים למימוש.
   א) LRU
   ב) אלגוריתם השעון
   ג) אלגוריתם הבוחר להחליף את הדף שלא נצטרך אותו הכי הרבה זמן בעתיד
   ד) אלגוריתם הבוחר להחליף דף שאינו חלק מה-working set, כלומר שלא השתמשנו בו ב-Δ הפקודות האחרונות, עבור Δ מתאים

4. איזו מהבעיות הבאות של user threads ניתנת בעצם לפתרון בספריה הממשת את ה-threads?
א) אי אפשר לנצל ריבוי של מעבדים או ליבות
ב) אם חוט אחד מבצע פעולת sleep (מבקש לא לרוץ זמן מסויים) כל החוטים בתהליך נחסמים
ג) אם חוט אחד מבצע פעולת I/O (למשל קריאה מקובץ) כל החוטים בתהליך נחסמים
ד) ביצוע של system call סובל מתקורה של trap

5. איזו מההצעות הבאות אינה שיטה אפשרית למניעת deadlock:
א) דרישה שכל תהליך ימנע מלבקש משאבים בצורה שתגרום ליצירת מעגל בגרף ההקצאות
ב) קביעת יחס סדר על המשאבים ודרישה שתהליך המחזיק במשאבים מסוימים יכול לבקש רק משאבים בעלי מספר גבוה יותר
ג) דרישה שכל תהליך יבקש את כל המשאבים שלו בבת אחת בתחילת הריצה
ד) דרישה שאם תהליך רוצה להשיג משאבים נוספים, הוא צריך קודם כל לשחרר את אלה שהוא כבר מחזיק

תהליך לא יכול לדעת בכלל על משאבים הנמצאים בידי תהליכים אחרים, וממילא לא יכול לחשב את הגרף

6. איזה מהמאורעות הבאים אינו גורם למעבר מידי להרצה של מערכת ההפעלה?
א) קריאה לפונקציה flush המסנכרנת מידע שנאגר בחוצצים (buffers) שונים לדיסק
ב) המשתמש מקיש על מקש enter במסוף המחשב (console)
ג) כרטיס הרשת מקבל הודעה מרשת התקשורת
ד) קריאה לפונקציה sin מהספריה המתמטית

7. ברצוננו להשתמש באותו מחשב עבור שני שרתי WWW שונים. כיצד ניתן לעשות זאת? (הנח שלא משתמשים בטכניקות של וירטואליזציה עליהן לא למדנו בקורס)
א) אין בכלל שום בעיה – פשוט מריצים שני תהליכים שמריצים את התכנה של שרת WWW
ב) אפשר לעשות זאת אם הם משתמשים במספרי port שונים
ג) אפשר לעשות זאת אם נותנים לכל תהליך כתובת IP שונה
ד) אי אפשר לעשות זאת

צריך רק שהלקוחות ידעו לאיזה port לפנות

8. התפקיד העיקרי של פרוטוקול IP הוא
א) לנתב מנות (הודעות) מרשת אחת לשניה באינטרנט
ב) לוודא שהמידע המשודר מגיע ליעדו בלי טעויות על ידי כך שמוסיפים לו מידע לבקרה (כמו parity)
ג) למנוע עומס יתר ברשת על ידי כך שמקטינים את קצב השידור כשיש עומס
ד) לתרגם כתובות "אנושיות" כגון cs.huji.ac.il לכתובות מספריות כגון 132.67.25.6

9. פרוטוקול TCP ישדר מנה (packet) שכבר שידר בעבר פעם נוספת אם
א) עבר הרבה זמן בלי שהתקבל עליה ack (כלומר היה timeout)
ב) התקבל ack עבור נתונים ששודרו מאוחר יותר
ג) הצד המקבל דיווח שהתפנה לו מקום בחוצצים (buffers)
ד) זה לא קורה כי TCP לא מבצע שידור חוזר של מנות

10. השרתים במערכת הקבצים המבוזרת NFS הם חסרי-מצב (stateless) כי
א) זה חוסך בזיכרון ומקטין את התקורה (overhead)
ב) זה מאפשר להתגבר בקלות על אובדן הקשר עם הלקוחות
ג) זה מאפשר להשתמש בפעולות אידמפוטנטיות (שניתן לחזור עליהן) על קבצים
ד) הם בכלל לא חסרי-מצב

11. האלגוריתם של פטרסון למניעה הדדית הוא (עבור שני תהליכים i=0,1)

```
going_in[i] = TRUE;
turn = 1 – i;
while (going_in[1-i] && turn==1-i)
    /* nothing */;
critical section
going_in[i] = FALSE;
```

איזו דרגה של הגינות הוא מבטיח?

א) התהליכים יכנסו לקטע הקריטי בסדר שבו ביצעו את ההשמה going_in[i] = TRUE
ב) מובטח כי בכל מקרה התהליכים יכנסו לקטע הקריטי לסירוגין (0,1,0,1,...)
ג) אם תהליך מחכה להכנס לקטע הקריטי, התהליך השני יכנס לפניו לכל היותר פעם אחת
ד) כשתהליך מחכה להכנס לקטע הקריטי, התהליך השני עלול להכנס מספר לא חסום של פעמים

12. נסמן את המיפוי משמות קבצים למספרי inode הנשמר בסיפריות על ידי "a" → 3 (כלומר קובץ a מיוצג במערכת על ידי inode מספר 3). איזה מהדגמים הבאים אינו יכול להתקיים במערכת קבצים?

א) בבלוק של ספריה אחת מופיע המיפוי "a" → 3 ובבלוק של ספריה אחרת מופיע "b" → 3
ב) בבלוק של ספריה אחת מופיע המיפוי "a" → 3 ובבלוק של ספריה אחרת מופיע "a" → 5
ג) בבלוק של ספריה מופיע המיפוי "a" → 3 ובבלוק אחר של אותה ספריה מופיע "b" → 3
ד) בבלוק של ספריה מופיע המיפוי "a" → 3 ובבלוק אחר של אותה ספריה מופיע "a" → 5

חלק ב' (45 נקודות)

ענה על 3 מתוך 4 השאלות הבאות.

1. i מחשב משרת רצף תהליכים המגיעים בהפרשי זמן של בדיוק דקה אחת. אורך כל תהליך בדיוק 45 שניות. התהליכים מקבלים שרות זה אחר זה, ללא הפקעה (preemption). הערך את הגדלים הבאים, או גבולות עליהם (למשל "גדול או שווה ל-13"):

ניצולת המעבד: 75%

הנצולת 75% בכל המקרים כי בכולם בממוצע כל דקה יש לנו 45 שניות של עבודה

זמן התגובה הממוצע: 45 שניות

ii. הערך את אותם גדלים, אם נתון שהתהליכים מגיעים בהפרשי זמן של דקה אחת בדיוק, אבל אורכם 45 שניות בממוצע, כאשר האורך מאופיין על ידי התפלגות יוניפורמית על הטווח של 30-60 שניות:

ניצולת המעבד: 75%

זמן התגובה הממוצע: 45 שניות

כיוון שאורך העבודות לא יותר מדקה, אין המתנה כלל, וזמן התגובה הממוצע שווה לזמן הריצה הממוצע

iii. הערך את הגדלים בשלישית, כאשר הפעם התהליכים מגיעים בהפרשי זמן של דקה בממוצע, ואורכם 45 שניות בממוצע, והתפלגויות הפרשי הזמן והאורכים הן ההתפלגות האקספוננציאלית:

ניצולת המעבד: 75%

זמן התגובה הממוצע: יותר מ-45 שניות (בעצם 3 דקות אם משתמשים בנוסחה, אבל זה לא נדרש)

מידי פעם יכול להיות תהליך ארוך והפרש קצר יותר, ואז יש המתנה וזמן התגובה מתארך

iv. ולבסוף, הערך את הגדלים כאשר התהליכים מגיעים בהפרשי זמן של דקה בממוצע, ואורכם 45 שניות בממוצע, אך הפעם ההתפלגויות הן התפלגות עם זנב כבד:

ניצולת המעבד: __75%___

בהתפלגות כזו ההסתברות לתהליך מאוד ארוך יותר גדולה, ולכן יש הסתברות רבה יותר להמתנה ארוכה

זמן התגובה הממוצע: _יותר מ-45 שניות (בעצם יותר מ-3 דקות)__

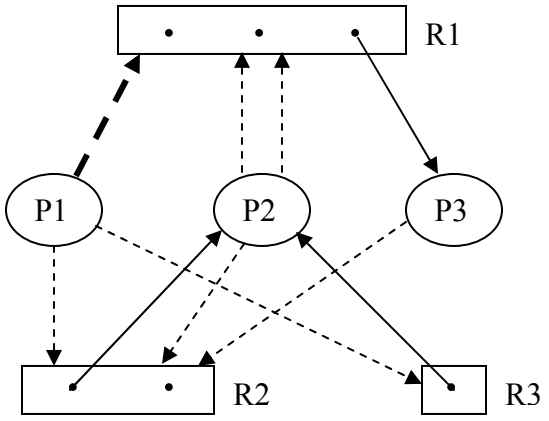
v. דרג את זמני התגובה הממוצעים הצפויים בארבעת הסעיפים לעיל (רשום i, ii, iii, ו-iv בסדר הנכון וביניהם הסימנים =, <, או ≤ לפי הצורך)

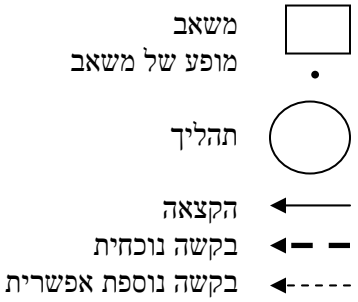
i = ii < iii < iv

vi. נתון כי זמני הריצה של תהליכים במערכת מבוזרת הם בעלי התפלגות עם זנב כבד. מה ההשלכות לגבי העומסים על המחשבים ברשת, כאשר כל מחשב מריץ עבודות של המשתמש שלו אך יכול גם להריץ עבודות של משתמשים אחרים? (סמן את כל הנכונות)

א) העומסים צפויים להיות שווים פחות או יותר, ואין צורך באיזון אקטיבי בין המחשבים
ב) יש לצפות שמספר קטן של מחשבים ברשת יהיו הרבה יותר עמוסים מהממוצע, ורצוי להעביר תהליכים מהם למחשבים אחרים
ג) שיטת איזון טובה היא להעביר כל תהליך חדש למחשב אחר מיד אחרי שהוא מתחיל לרוץ
ד) שיטת איזון טובה היא לבחור שני מחשבים באופן אקראי ולהעביר עומס עודף לפחות עמוס מביניהם
ה) עדיף לא להעביר תהליכים שכבר רצו הרבה זמן כי הם צפויים להסתיים בקרוב

2. אלגוריתם הבנקאי משתמש בגרף הקצאות משאבים כדי לבחון אם הקצאה חדשה תשאיר את המערכת במצב בטוח. עיין בגרף ההקצאות הנתון כאן וענה על השאלות.





i. האם ניתן לספק את הבקשה הנוכחית של P1 למשאב R1, כלומר האם סיפוק הבקשה יוביל למצב בטוח?

כן [x] לא [ ]

אם כן, איזה סדר הרצה מדגים זאת? ראשון: _P3__ שני: _P2__ שלישי: _P1__

האם יש סדר נוסף אפשרי? רשום אותו או "לא": ראשון: _____ שני: _____ שלישי: _____

ii. נניח שסיפקנו את הבקשה הנוכחית והקצנו מופע של R1 לתהליך P1, וכעת הוא מבקש לקבל גם מופע של משאב R2. האם ניתן לספק בקשה זאת ?

א) אסור ל-P1 לבקש הקצאה כזאת בכלל
ב) המצב היה לא בטוח מראש בגלל ההקצאה הקודמת, ולכן השאלה לא רלוונטית
ג) לא, זה יוביל למצב לא בטוח
ד) כן, זה היה מצב בטוח ויוביל למצב בטוח חדש

iii. לחילופין, נניח שסיפקנו את הבקשה הנוכחית והקצנו מופע של R1 לתהליך P1, וכעת תהליך P3 מבקש לקבל מופע של משאב R3. האם ניתן לספק בקשה זאת ?

א) אסור ל-P3 לבקש הקצאה כזאת בכלל
ב) המצב היה לא בטוח מראש בגלל ההקצאה ל-P2, ולכן השאלה לא רלוונטית
ג) לא, זה יוביל למצב לא בטוח
ד) כן, זה היה מצב בטוח ויוביל למצב בטוח חדש

3. קטע הקוד הבא משתמש ב-setjmp ו-longjmp כדי להעביר שליטה בין פונקציות, בדומה לספרית החוטים של תרגיל 2. ניתן להניח שההגדרות נכונות, ושאין צורך בתרגום כתובות כפי שנעשה בתרגיל.

```
1     #include <setjmp.h>
2     #define STACK_SIZE 4096
3     char stack[STACK_SIZE];
4     jmp_buf jbuf[2];

5     void f()
6     {
7        int i=0;
8        setjmp(jbuf[0]);
9        printf("in f (%d)\n",i++);
10       longjmp(jbuf[1],1);
11    }

12    int main()
13    {
14       (jbuf[0]->__jmpbuf)[SP] = stack + STACK_SIZE - sizeof(int);
15       (jbuf[0]->__jmpbuf)[PC] = f;
16       jbuf[0]->__mask_was_saved = 1;

17       setjmp(jbuf[1]);
18       printf("in main\n");
19       sleep(1);

20       longjmp(jbuf[0],1);
21       return 0;
22    }
```

i. מה יקרה כאשר נריץ תכנית זו?

א) התכנית תסיים בלי להדפיס שום פלט
ב) התכנית תדפיס פלט ואז תסיים
ג) התכנית תדפיס פלט באופן אינסופי ולא תסיים
ד) התכנית תגרום ל-exception ותהרג ע"י מערכת ההפעלה מבלי להדפיס פלט
ה) התכנית תדפיס פלט ואז תגרום ל-exception ותהרג ע"י מערכת ההפעלה

אם התכנית תדפיס פלט, מהו הפלט שיודפס?

```
in main
in f(0)
in main
in f(1)
…
```

כעת נשנה את הפונקציה main באופן הבא:

```
12  int main()
13  {
14      (jbuf[0]->__jmpbuf)[SP] = stack + STACK_SIZE - sizeof(int);
15      (jbuf[0]->__jmpbuf)[PC] = f;
16      jbuf[0]->__mask_was_saved = 1;
17      longjmp(jbuf[0],1);

18      setjmp(jbuf[1]);
19      printf("in main\n");
20      sleep(1);

21      return 0;
22  }
```

ii. מה יקרה כאשר נריץ את התכנית בגרסה זו?

א) התכנית תסיים בלי להדפיס שום פלט
ב) התכנית תדפיס פלט ואז תסיים
ג) התכנית תדפיס פלט באופן אינסופי ולא תסיים
ד) התכנית תגרום ל-exception ותהרג ע"י מערכת ההפעלה מבלי להדפיס פלט
ה) התכנית תדפיס פלט ואז תגרום ל-exception ותהרג ע"י מערכת ההפעלה

אם התכנית תדפיס פלט, מהו הפלט שיודפס?

```
in f(0)
```

iii. האם ניתן לשנות את התכנית המקורית על ידי <u>מחיקת שורות בלבד</u> כדי לקבל פלטים כאלה? (סמן את כל האפשרויות הנכונות)

א) פלט ריק (0 שורות)
ב) פלט של שורה אחת בלבד
ג) פלט של בדיוק שתי שורות
ד) פלט של בדיוק 3 שורות
ה) פלט של בדיוק 4 שורות

4. איך מממשים את הפעולות הבאות במערכת הקבצים של מערכת Unix קלאסית?

i. הפקודה mv "מזיזה" קובץ ממקומו המקורי למקום אחר בהיררכית הקבצים. כדי לממש את הפקודה mv /a/b/c /d/e צריך לבצע (בין היתר) את הפעולות הבאות (סמן את כולן):
   א) ליצור קובץ חדש, להקצות לו inode, ולמפות אותו לשם e בספריה d
   ב) להעביר את הבלוקים של הקובץ /a/b/c ל-inode החדש
   ג) להעביר את מיפוי ה-inode של הקובץ מ-/a/b ל-/d

ii. הפקודה open מקבלת את השם המלא של קובץ (ה-path, כלומר המסלול מהשורש עד הקובץ), ופותחת אותו לגישה. לשם כך צריך לבצע (בין היתר) את הפעולות הבאות (סמן את כולן):
   א) לקרוא את ה-inodes של כל הספריות במסלול
   ב) לבדוק הרשאויות גישה לכל הספריות במסלול
   ג) להקצות file descriptor לכל ספריה במסלול

iii. הפקודה seek מעדכנת את המצביע לקובץ פתוח כך שפעולת read או write עוקבת תתבצע במקום המצויין. לשם כך צריך לבצע (בין היתר) את הפעולות הבאות (סמן את כולן):
   א) לעדכן את המצביע בטבלת הקבצים הפתוחים
   ב) לעדכן את המצביע ב-inode של הקובץ
   ג) לקרוא את הבלוק המתאים מהקובץ ל-buffer cache

iv. הפקודה write כותבת נתונים לתוך קובץ פתוח. לשם כך צריך לבצע (בין היתר) את הפעולות הבאות (סמן את כולן):
   א) לקרוא את הבלוק מהדיסק בכדי לעדכן אותו, אם הפעולה היא על בלוק שלם ומעלה
   ב) לעדכן את המצביע המציין את מיקום הפעולה הבאה על הקובץ
   ג) לעדכן את ה-inode אם הוקצה לקובץ בלוק נוסף

הניסוח של א' בעייתי והסעיף בוטל

רק לבלוקים הראשונים מעדכנים את ה-inode, אח"כ זה בלוקים לא ישירים

v. הפקודה read קוראת נתונים מקובץ פתוח אל חוצץ של התכנית הקוראת. לשם כך צריך לבצע (בין היתר) את הפעולות הבאות (סמן את כולן):
   א) לעדכן את ה-inode של הקובץ לגבי זמן הגישה האחרונה
   ב) לוודא שיש למשתמש הזכות לקרוא את הקובץ
   ג) להעתיק את הקובץ ל-buffer cache

בדיקת הרשאויות גישה נעשית ב-open, לא ב-read

מעתיקים רק את הבלוקים הנחוצים לגישה הזו