אוניברסיטת בן-גוריון בנגב, המחלקה למדעי המחשב

מועד ב' במערכות הפעלה

תאריך הבחינה: 15 ביולי, 2019
שם הקורס: מערכות הפעלה
מספר הקורס: 20213031
שנה: 2019, סמסטר: ב', מועד: ב'
משך הבחינה: שלוש שעות
חומר עזר: אסור

מרצים: איתי דינור, דני הנדלר ומרינה קוגן-סדצקי.

מתרגלים: אור דינרי, מתן דרורי, סימיון נוביקוב,

צחי ספורטה, עמית פורטנוי וירין קופר.

ענו על כל השאלות: סה"כ 100 נקודות.

1. ניהול זיכרון (25 נקודות)

א. **(16 נקודות)** בסעיף זה נתייחס למערכת הפעלה התומכת בזיכרון וירטואלי המנוהל במנגנון paging. גודל דף במערכת הוא 8 KB. המערכת תומכת בשתי רמות של טבלאות דפים, כאשר ברמה הראשונה יש טבלה אחת. גודל כל טבלת דפים (בכל רמה שהיא) הוא דף אחד (כלומר 8 KB) וגודל כניסה בכל טבלת דפים הוא 4 בתים. הקצאות הזיכרון במערכת הן תמיד בכפולות של 8 KB.

ענו על השאלות הבאות ונמקו את תשובותיכם בקצרה.

i. **(6 נקודות)** מהו הגודל המקסימלי של הזיכרון הווירטואלי שהמערכת יכולה לתמוך בו?

ii. **(6 נקודות)** נתון תהליך משתמש אשר מנצל 68 MB (68 מגה-בית) ממרחב הזיכרון הוירטואלי שלו וכולם נמצאים בזיכרון הפיסי. מהו המספר המינימאלי של טבלאות דפים הדרוש לתהליך (בשתי הרמות)?

iii. **(4 נקודות)** בהמשך ל-ii, נתון שמערכת ההפעלה ביצעה swap out לדיסק ל 32 MB של זיכרון מתוך ה 68 MB של התהליך. ענו על אותה שאלה לאחר ביצוע ה swap out.

ב. **(9 נקודות)** נתונה מערכת הפעלה התומכת בזיכרון וירטואלי המנוהל במנגנון paging. בנוסף נתון כי מערכת ההפעלה רצה על מחשב עם מעבד יחיד ורשומות ה TLB במעבד <u>אינן</u> מכילות שדה של מזהה תהליך. נאמר שפעולה מסוימת <u>מחייבת שינוי</u> של ה TLB, אם קיים תרחיש שבו ביצוע הפעולה ללא שינוי ה TLB גורר פגיעה בנכונות תוכנית משתמש. עבור כל אחת מהפעולות מבין הפעולות הבאות, רשמו האם היא מחייבת שינוי של ה TLB עבור המערכת הנתונה. נמקו את תשובותיכם בקצרה.

i. **(5 נקודות)** מיפוי (הקצאה) של דף חדש בזיכרון הווירטואלי של התהליך הנוכחי (התהליך הרץ על המעבד) לדף בזיכרון הפיסי.

ii. **(4 נקודות)** ביצוע swap out לדיסק עבור דף הממופה לזיכרון הווירטואלי של התהליך הנוכחי.

2. חוטים ותהליכים (25 נקודות)

נתונה התוכנית הבאה. תכנית זו יוצרת שני חוטים. החוט הראשון מבצע את הפונקציה f1 והחוט השני מבצע את הפונקציה f2. החוט הראשי ממתין לסיום של שני החוטים הללו לפני שהוא מבצע את שתי פקודות ההדפסה. לאורך כל השאלה יש להניח שכל הקריאות ל-thread_create ו-thread_join ו-fork מצליחות. **שימו לב: הסעיפים הבאים אינם קשורים זה לזה. כל השינויים בסעיפים אלו מתייחסים לתוכנית המקורית המופיעה להלן.**

```
// Global variables
int g1 = 0, g2 = 0;

f1( )
{ g2 = g1 + 1; }

f2( )
{ g1 = g2 – 1; }

main( )
{
    thread_create(f1);
    thread_create(f2);

    // Wait for all threads to terminate
    while (thread_join( ) == 0);

    printf("g1 = %d", g1);
    printf("g2 = %d", g2);
}
```

א. (**6 נק'**) מה הם הפלטים האפשריים של התוכנית? עבור כל פלט אפשרי, כתבו מהו והסבירו בקצרה.

ב. (**6 נק'**) הוחלט להחליף את הפונקציה main בפונקציה הבאה. כיצד ישתנה הפלט של התוכנית (אם בכלל)? הסבירו בקצרה, התייחסו לכל התוצאות האפשריות.

```
main( )
{
    if (fork() == 0)
        {
         f1();
         return;
        }
    if (fork() == 0)
        {
         f2();
         return;
        }
    while (wait( ) == -1); // Wait for all child processes to terminate
    printf("g1 = %d", g1);
    printf("g2 = %d", g2);
}
```

ג. (**6 נק'**) הוחלט לשנות את התוכנית המקורית ע"י הוספת שני משתנים גלובליים מסוג mutex ולשנות את הפונקציות f2 ו-f2 כלהלן. כיצד ישתנה הפלט של התוכנית (אם בכלל)? הסבירו בקצרה, התייחסו לכל התוצאות האפשריות.

```
mutex m1, m2 // Two additional global variables
f1( )
{
        lock(m1);
        g2 = g1 + 1;
        unlock(m1);
}
f2( )
{
        lock(m2);
        g1 = g2 – 1;
        unlock(m2);
}
```

ד. (**7 נק'**) הוחלט לשנות את התוכנית המקורית ע"י הוספת שני משתנים גלובליים מסוג mutex ולשנות את הפונקציות f2 ו-f2 כלהלן. כיצד ישתנה הפלט של התוכנית (אם בכלל)? הסבירו בקצרה, התייחסו לכל התוצאות האפשריות.

```
mutex m1, m2 // Two additional global variables
f1( )
{
        lock(m1);
        lock(m2);
        g2 = g1 + 1;
        unlock(m2);
        unlock(m1);
}
f2( )
{
        lock(m2);
        lock(m1);
        g1 = g2 – 1;
        unlock(m1);
        unlock(m2);
}
```

3. סינכרוניזציה (25 נקודות)

א. **(12 נק')** להלן קוד המממש סמאפור בינארי באמצעות מוניטור.

```
BinarySemphore: Monitor {

 1 condition variable cv
 2 int free initially 1
 3 void down( ) {
 4    while (free ≤ 0)
 5       { cv.wait }
 6    free--
 6 }
 7 void up( ) {
 8    if (free < 1) {
 9       free++
10       cv.signal
11    }
12 }
```

i. **(6 נק')** בהינתן שהמוניטור הוא מסוג Hoare, האם המימוש הנ"ל מקיים את הסמנטיקה של סמאפור בינארי? אם כן הסבירו בקצרה, אחרת תארו תסריט מדויק המהווה דוגמא נגדית.

ii. **(6 נק')** ענו על אותה שאלה בהינתן שהמוניטור אינו מסוג Hoare.

ב. **(13 נק')** להלן אלגוריתם למניעה הדדית.

```
shared int turn
shared boolean busy initially false
Program for process i ∈ (1,…,n)
1 do {
2     do {
3         turn := i
4         } while (busy)
5     busy := true
6     } while (turn != i)
7 Critical section
8 busy := false
```

i. **(6 נק')** האם האלגוריתם מקיים מניעה הדדית? אם כן, נמקו בקצרה. אם לא, כתבו תסריט מדויק המדגים כי יש הפרה של מניעה הדדית.

ii. **(7 נק')** האם האלגוריתם מקיים חופש מקיפאון? אם כן, נמקו בקצרה. אם לא, כתבו תסריט מדויק המדגים כי אין חופש מקיפאון.

4. ניהול קבצים, וירטואליזציה (25 נקודות)

בכל הסעיפים בשאלה זו, יש לנמק במדויק ובקצרה.

א. **(5 נק')** במערכת הפעלה UNIX נמחק התוכן של קובץ הספרייה usr/ שהכילה קבצים רגילים בלבד. ידוע שבספריות אחרות על הדיסק, קיים לכל אחד מקבצים אלה גם soft link וגם hard link. האם ניתן על סמך links אלו לשחזר את תוכנו של קובץ הספרייה?

ב. נניח כי בזמן נתון, במערכת הפעלה UNIX, קיים קובץ file/ וקיימים X inodes פנויים.

i. **(3 נק')** זרובבל יוצר hard link חדש לקובץ file/ , כמה inodes פנויים יש במערכת כעת?

ii. **(3 נק')** ענו על אותה שאלה בהנחה שזרובבל יוצר soft link חדש.

ג. **(5 נק')** מהו קובץ דליל (sparse file)?

ד. **(4 נק')** הסבירו בקצרה מהי privileged instruction ומהי sensitive instruction.

ה. **(5 נק')** הסבירו בקצרה מהי וירטואליזציה בשיטת trap and emulate ומדוע היא לא הייתה אפשרית בארכיטקטורות ישנות של x86, בהן לא הייתה תמיכה של החומרה בוירטואליזציה.

בהצלחה!