מבחן במערכות הפעלה

מועד א' תשע"ו **עם פתרון**

אוניברסיטת חיפה, החוג למדעי המחשב

מורה: רחל קולודני
מתרגל: רמי עילבוני

חלק ראשון: 30 נקודות

קבע/י לגבי כל אחת מהאמירות הבאות אם היא אמת או שקר. עבור כל תשובה נכונה תקבלו 3+ נקודות, עבור כל תשובה שגויה תקבלו 1- נקודות

תהליך שהסתיים אבל ההורה שלו עדיין לא קרא ל-()wait נקרא תהליך זומבי (zombie)	**נכון** / לא נכון
קריאת המערכת ()exec יוצרת תהליך חדש	נכון / **לא נכון**
Thrashing לא יכול להיות בעיה במחשב עם זיכרון פיסי של 10GB	נכון / **לא נכון**
החלקים בזיכרון המוקצים בעזרת buddy system הם כולם באותו הגודל	נכון / **לא נכון**
אם משתמשים באלגוריתם FIFO להחלפת דפים בזיכרון הפיסי, ואם מגדילים את מספר הדפים בזיכרון זה (כלומר, יותר page frames), מובטח שיהיו פחות page faults	נכון / **לא נכון**
במערכות מחשב עם זיכרון וירטואלי, גודל הכתובת של הזיכרון הוירטואלי וגודל הכתובת של הזיכרון הפיסי תמיד אותו הדבר	נכון / **לא נכון**
לא יכול להיות deadlock אם אסור לתהליך לבקש משאב בזמן שהוא מחזיק משאב אחר	**נכון** / לא נכון
יכולה להיות החטאה של TLB (TLB miss) גם אם הדף נמצא בזיכרון הראשי	**נכון** / לא נכון
כדי להשתמש בחוטים (threads) ב-Java, אפשר לממש את הממשק (interface) Runnable	**נכון** / לא נכון
בלינוקס שלמדנו בתרגולים ובגלל מנגנון copy-on-write, כל גישה לכתיבה מצד האבא/הבן תגרום ל-page-fault	נכון / **לא נכון**

חלק שני: 72 נקודות

סמנ/י את התשובה הנכונה לכל אחת מהשאלות. עבור כל תשובה נכונה תקבלו +4 נקודות, עבור כל תשובה שגויה תקבלו 1- נקודות

<u>שאלה 1:</u>

תהליך אב מריץ את תכנית א', עד לנקודה בה תכנן לקרוא ל-()fork ואז ל-()exec כך שתהליך הבן יריץ את תכנית ב'. מה יקרה אם סדר קריאות המערכת (system call) יוחלף בטעות ויקראו קודם ל-()exec ורק אח"כ ל-()fork?

(א) האב יריץ את תכנית א', והבן יריץ את תכנית ב'
(ב) האב יריץ את תכנית ב', והבן יריץ את תכנית א'
(ג) רק תכנית א' תרוץ בשני התהליכים
(ד) רק תכנית ב' תרוץ בשני התהליכים
(ה) אף אחת מהתשובות אינה נכונה

שאלה 2:

כאשר שני תהליכים מתקשרים ביניהם (inter process communication), אם גם ה-()send וגם ה-()receive הם blocking, אזי התקשורת נקראת:

(א) תקשורת סינכרונית synchronous message
(ב) תקשורת א-סינכרונית asynchronous message
(ג) רנדוו Rendezvous
(ד) תקשורת חסומה blocked message
(ה) אף אחת מהתשובות אינה נכונה

שאלה 3:

התהליך מריץ את התכנית הבאה:

```
include<stdio.h>

int A,B;

int Add()
{
        return A + B;
}

int main()
{
        int answer;
        A = 5;
        B = 7;
        answer = Add();
        printf("%d\n",answer);
        return 0;
}
```

איפה נשמרים A ו-answer בזיכרון של התהליך?

(א) A ו-answer במחסנית (stack)
(ב) A ו-answer בערימה (heap)
(ג) A ו-answer באזור הדטה (data section)
(ד) A במחסנית ו-answer באזור הדטה
(ה) A באזור הדטה ו- answer במחסנית
(ו) A במחסנית ו-answer בערימה
(ז) אף אחת מהתשובות אינה נכונה

שאלה 4:

(השלימו) _____ היא נוסחה שמזהה את הפוטנציאל לשיפור הביצועים מהוספת ליבות להרצת חישוב שיש בו חלקים מקבילים וסדרתיים

(א) מיקבול תהליכים (task parallelism)
(ב) מיקבול נתונים (data parallelism)
(ג) פיצול נתונים (data splitting)
(ד) חוק אמדל (Amdahl's law)
(ה) אף תשובה לא נכונה

שאלה 5:

כשתהליך אב מריץ תהליך בן, איזה מהבאים יכול להתקיים:

(א) תהליך הבן רץ במקביל לתהליך האב
(ב) תהליך הבן מריץ תכנית שונה מתהליך האב
(ג) תהליך הבן הוא עותק של תהליך האב
(ד) כל הסעיפים א-ג
(ה) אף אחד מהסעיפים אינו נכון

שאלה 6:

הרעבה מתארת מצב בו חוט (thread):

(א) רץ בלולאה אינסופית עד שנגמר לו הזיכרון
(ב) ה-scheduler לא מריץ אותו לעולם
(ג) לא מצליח לקחת מנעול (lock) שצריך לקטע קוד קריטי
(ד) לא מצליח לקרוא לפקודת fork()
(ה) לא מצליח לקבל הקצאה של זיכרון

שאלה 7:

במערכת מחשב scheduler שהוא מבוסס עדיפויות לפי הטבלה שלפניכם (עדיפות 1 היא הגבוהה ביותר) ו-pre-emptive. התהליכים P1,…,P5 מגיעים לתור ה-ready, במרווחים של 10 מילי-שניות, ולפי הסדר: 5, ואז 3, ואז 4, ואז 2, ואז 1.

תהליך	זמן נידרש (מילי-שניות)	עדיפות
P1	40	1
P2	10	2
P3	20	3
P4	30	4
P5	20	5

מתי (באיזה זמן) יסתיים התהליך P5? **120**

שאלה 8:

במערכת מחשב scheduler שהוא מבוסס עדיפויות לפי הטבלה שלפניכם (עדיפות 1 היא הגבוהה ביותר) ו-pre-emptive. תהליכים P1 ו-P4 מגנים על משאב משותף בעזרת סמפור S. תהליך P4 מתחיל לרוץ בזמן t=0, אחרי שלקח את S (הניחו שהתהליך לא ישחרר את S עד שיסיים את זמן הריצה הנדרש שלו).

בזמן t=0, תהליך P4 מוכן לריצה
בזמן t=10, תהליך P1 מוכן לריצה
בזמן t=20, תהליך P2 מוכן לריצה
בזמן t=40, תהליכים P3 מוכן לריצה

תהליך	זמן נידרש (מילי-שניות)	עדיפות
P1	40	1
P2	10	2
P3	20	3
P4	30	4

(א) זוהי דוגמא של היפוך עדיפויות (priority inversion) בו תהליכים עם עדיפות נמוכה יותר (P2, P3) מעקבים תהליך עם עדיפות גבוהה יותר P1

(ב) זוהי דוגמא של היפוך עדיפויות (priority inversion) בו תהליך עם עדיפות נמוכה יותר (P4) מעקב תהליך עם עדיפות גבוהה יותר P1

(ג) התהליכים ירוצו על פי סדר העדיפויות שלהם וזמני ההגעה: P4, ואז P1, ואז P2, P3, ושוב P4.

(ד) זוהי דוגמא של היפוך עדיפויות נדיר (rare priority inversion)בו רק תהליך אחד עם עדיפות נמוכה יותר מעקב תהליך עם עדיפות גבוהה יותר P1

(ה) אף אחת מהתשובות אינה נכונה

שאלה 9:

נתונה מערכת מחשב עם טבלת דפים ב-2 רמות (two-level page table) ששמורה בזכרון הראשי. ידוע שגישה אחת לזיכרון הראשי לוקחת 200 ננו-שניות

(א) כמה זמן יקח לגשת למשתנה שנמצא בזכרון הראשי ?

200+200+200 = 600 ns

(ב) נניח ונוסיף TLB כך שבסיכוי של 75% הגישה לזיכרון היא כתובת שמתוארת ב-TLB, ונניח גם שהגישה לטבלת ה-TLB עצמה לוקחת זמן 0 אם הכניסה באמת בטבלה. מהו זמן הגישה האפקטיבי לזיכרון (effective memory reference time) ?

0.75*200+0.25*600 = 300 ns

שאלה 10:

אפשר לממש את רשימת המקום החופשי בדיסק (free-space list) בעזרת ווקטור של ביטים (bit vector). מה מהבאים הוא חסרון של השיטה הזו?

(א) כדי לעבור על הרשימה, צריך לקרוא כל בלוק מהדיסק
(ב) עבור דיסקים גדולים, אי אפשר לשמור את כל הרשימה בזכרון הראשי
(ג) השיטה יותר מסובכת למימוש משיטות אחרות
(ד) עבור דיסקים קטנים, זו שיטה לא אפשרית
(ה) אף אחת מהתשובות אינה נכונה

שאלה 11:

ברוב מערכות ההפעלה ה-general purpose השיטה לקשירת (binding) הוראות לכתובות זיכרון היא:

(א) בזמן אינטרפטים: interrupt binding
(ב) בזמן קומפילציה: compile-time binding
(ג) בזמן טעינה: load time binding
(ד) בזמן לינק: link-time binding
(ה) בזמן ריצה: execution time binding

שאלה 12:

מה צריכים להוסיף למערכת מחשב שמממשת time sharing שלא היה ממומש במערכת שבה יש רק תהליכים מרובים (multiprogramming) ?

(א) מנגנון של trap
(ב) היכולת להריץ קטעי קוד עם הרשאות של kernel
(ג) פרוסות זמן קצרות יותר
(ד) שעון עצר שמייצר trap במרווחי זמן קבועים
(ה) מימוש של מבנה נתונים של PCB

שאלה 13:

נתונים שני תהליכים ב-Linux המריצים את קטע הקוד הבא (x משתנה גלובלי שערכו 0).

```
printf("%d", &x);
```

(1) יתכן שלשני התהליכים יש אותו PID
(2) תוצאת ההדפסה של שני התהליכים תמיד תהיה אותו הערך

(א) (1) נכון –(2) נכון
(ב) (1) נכון ו-(2) לא נכון
(ג) (1) לא נכון ו-(2) נכון
(ד) (1) לא נכון ו-(2) לא נכון

שאלה 14:

במערכת נתונה עם מרחב כתובות של 32 ביט עם דפים בגודל 4KB (kilo BYTE) משתמשים בזיכרון ראשי (RAM) בנפח 16MB. טבלת הדפים היא ברמה אחת שמוקצת תמיד ב-RAM, כל כניסה בה מכילה מספר מסגרת (frame number) ו-4 ביטי בקרה.

מהו גודל הזיכרון הנדרש ב-MB (mega byte) לאחסון טבלת הדפים עבור תהליך בודד?

א. 5 MB
ב. 4 MB
ג. 2 MB
ד. 3 MB
ה. 1 MB

שאלה 15:

בתרגול למדנו שהמימוש בלינוקס של אזור זיכרון משותף בין שני תהליכים עם process descriptors PD1, PD2 שרצים ב- user space נעשה על ידי:

א. הצבעה אל האזור המשותף מ מתארי מרחבי הזיכרון (memory descriptor) של שני התהליכים אליהם מצביעים PD1, PD2
ב. הגדרת טבלאות התרגום כך שחלק מהכתובות הליניאריות בשני התהליכים מתורגמות לאותן הכתובות פיסיות
ג. הזזת אזור הזיכרון המשותף לאזור הזיכרון של הגרעין (kernel) ואשר אליו גישה מכל טבלאות התרגום של כל התהליכים
ד. להצביע מ-PD1, PD2 למרחב זיכרון נוסף שמצביע אל האזור המשותף
ה. אי אפשר לשתף אזור זיכרון בין כמה תהליכים

שאלה 16:

אם ידוע שקריאת המערכת pthread_create הצליחה אבל החוט לא רץ לעולם, מה נכון:

א. החוט מחכה על מנעול
ב. החוט הראשי בתהליך זה סיים על ידי קריאה ל- pthread_exit
ג. חוט אחר באותו התהליך ביצע פעולה לא חוקית
ד. הוא כן רץ אבל בתהליך אחר
ה. לא עשינו pthread_mutex_init

שאלה 17:

מהו גודל הזיכרון הפיסי הנתמך במערכת הפעלה עם paging בו יש טבלת דפים (page table) המכילה 64 כניסות (entries), בה כל אחת באורך 11 ביטים (כולל ביט של valid/invalid), וגודל דף של 512 bytes ?

(א) 2^{11} bytes
(ב) 2^{19} bytes
(ג) 2^{15} bytes
(ד) 2^{25} bytes
(ה) אף אחת מהתשובות אינה נכונה

שאלה 18:

לפניכם שתי פונקציות למימוש הגנה על קטע קוד קריטי בעזרת הפעולה האטומית test-and-set ומשתנה משותף X שמאותחל ל-0. לפני הכניסה לקטע הקריטי יש לקרוא ל-enter_CS ולאחריו ל-leave_CS

```
void enter_CS(X)
{
    while test-and-set(X) ;
}
void leave_CS(X)
{
   X = 0;
}
```

(א) **בפתרון הנ"ל לבעיית קטע הקוד הקריטי לא יכול להיות deadlock**

(ב) בפתרון הנ"ל לבעיית קטע הקוד הקריטי **לא** יכולה להיות הרעבה (starvation)

(ג) בפתרון הנ"ל לבעיית קטע הקוד הקריטי **לא** יכול להיות deadlock **ולא** יכולה להיות הרעבה (starvation)

(ד) התהליכים הקוראים לפונקציות הנ"ל נכנסים לקטע הקוד הקריטי בסדר FIFO

(ה) יותר מתהליך אחד יכול להיכנס לקטע הקוד הקריטי בו זמנית

בהצלחה !!!