האוניברסיטה הפתוחה

מס' שאלון - 513

1

כ"ד בשבט תשע"ג

82

מספר התלמיד הנבחן

רשום את כל תשע הספרות

בפברואר 2013

מס' מועד	סמסטר 2013א
	20594 / 4

שאלון בחינת גמר

- 20594 מערכות הפעלה

משך בחינה: 3 שעות

בשאלון זה 11 עמודים

מבנה הבחינה:

קראו בעיון לפני שתתחילו בפתרון הבחינה!

- א. בבחינה זו 20 שאלות סגורות (מבחן אמריקאי). עליכם לבחור בכל פעם בתשובה יחידה מבין התשובות המוצעות ולהקיף בעיגול את אות התשובה שבחרתם, על גבי השאלון עצמו בלבד ליד טקסט השאלה.
 - ב. הציון נקבע על פי מספר התשובות הנכונות. כל תשובה נכונה מקנה 5 נקודות.
- ג. במקרה של ספק בהבנת ניסוח השאלה, ניתן לכתוב על גבי שאלון הבחינה הערה המבהירה את דרך הבנתכם את השאלה. אין לכתוב נימוקים לתשובה!

	חומר עזר:
	כל חומר עזר אסור בשימוש, פרט למחשבון.
בהצלחה !!!	
	החזירו למשגיח את השאלון וכל נוזר אחר שדוכלתם בתוב מחברת בתשובות

: (SIG IGN אשר אי-אפשר להתעלם ממנו (באמצעות) בחרו סיגנל (signal) בחרו

- SIGINT .א
- SIGKILL .2
- SIGSEGV .x
- SIGALRM .7

שאלה 2

איזו מבין התופעות הבאות בהכרח קשורה למקביליות בין תהליכים/תהליכונים!

- (deadlock) א. קיפאון
- ב. הרעבה (starvation)
- ג. סחרור בזיכרון (threshing)
- ד. כל התשובות הקודמות הן נכונות

שאלה 3

מערכת הפעלה יוניקס מנסה לתת לתהליכים בעלי עדיפות גבוהה timeslice ארוך, ואילו לתהליכים בעלי עדיפות נמוכה מקצה timeslice קצר. תהליכים בעלי עדיפות גבוהה הם בדרך-כלל התהליכים בעלי עדיפות נמוכה מקצה bounded IO), ואילו תהליכים בעלי עדיפות נמוכה הם התהליכים המבצעים פעולות קלט/פלט (bounded CPU). כתוצאה מכך, במקרים שכל התהליכים במערכת יוניקס הם תהליכים חישוביים, יקרו הרבה החלפות תוכן (context switches). כמובן שהדבר לא רצוי, מכיוון שהיינו רוצים לתת לתהליכים חישוביים לרוץ כמה שיותר זמן וללא החלפות תוכן תדירות. בחרו שיטת תזמון הפותרת בעיה זו:

- first come first served FIFO .א.
- shortest remaining time next
 - shortest job first λ
 - fair share scheduling .7

האם השימוש בזיכרון מטמון (Buffer Cash) מסוג write-trough מסוג האם השימוש בזיכרון מטמון האם של מערכת יחסינותה של מערכת הקבצים בהשוואה לשימוש בזיכרון מטמון מסוג

- א. כן. וזאת עקב ביצוע מהיר יותר של כתיבות בלוקים לדיסק.
 - ב. לא. וזאת עקב העיכוב בעדכון בלוקים בדיסק.
- ג. לא. כוון שחסינותה של מערכת הקבצים איננה מושפעת כלל וכלל מזיכרון מטמון.
 - ד. אף תשובה קודמת איננה נכונה.

הערה: מערכת קבצים חסינה היא מערכת קבצים שלגביה קיים סיכוי קטן להישאר במצב של חוסר עקביות.

שאלה 5

שני תהליכונים של אותו תהליך מסתנכרנים באמצעות סמפור קרנל (סמפור המסופק עייי גרעין של מערכת הפעלה). בחרו טענה נכונה:

- א: אם שני התהליכונים הללו הם תהליכונים ברמת משתמש (user threads), אזי ייתכן קיפאון א. אם שני התהליכונים הללו הם תהליכונים ברמת משתמש (deadlock).
- ב. אם שני התהליכונים הללו הם תהליכונים ברמת משתמש (user threads), אזי קיפאון לא ייתכן.
- ג. אם שני התהליכונים הללו הם תהליכונים ברמת קרנל (kernel threads), אזי קיפאון לא ייתכן.
- ד. אם שני התהליכונים הללו הם תהליכונים ברמת קרנל (kernel threads), אזי הרעבה ד. אם שני התהליכונים הללו הם תהליכונים לא תיתכן.

שאלה 6

במערכת עם ניהול זיכרון באמצעות דפדוף (paging) נתון כי:

כתובת מדומה (virtual address) הינה בת

64 Kbytes גודל הדף הוא

4 bytes גודל המילה הוא

מהי כמות הדפים המכסימלית בזיכרון המדומה!

- 2^40 א.
- ב. 2^50
- د. 2^54
- 2^60 .ד

בחרו מה תהיה התוצאה של הרצת הקוד:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdlib.h>
main() {
  char fn[]="readlink.file";
  char sl[]="readlink.symlink";
  char buf[30];
  int file descriptor;
  if ((file_descriptor = creat(fn, S_IWUSR)) < 0)</pre>
    perror("creat() error");
  else {
    close(file_descriptor);
    if (symlink(fn, sl) != 0)
      perror("symlink() error");
    else {
      if (readlink(sl, buf, sizeof(buf)) < 0)</pre>
        perror("readlink() error");
      else printf("readlink() returned '%s'
      for '%s'\n", buf, sl);
      unlink(sl);
    unlink(fn);
}
```

'readlink.file' for 'readlink.symlink'	۸.
readlink() returned 'readlink.file'	ב.
readlink() returned 'readlink.file' for 'readlink.symlink'	
'readlink.symlink'	٦.

: sigaction -בחרו טענה נכונה לגבי

- א. אפשר באמצעותה לתפוס או להתעלם מהסיגנל SIGKILL.
- ב. אפשר באמצעותה רק לתפוס אך לא להתעלם מהסיגנל SIGKILL
- ג. אפשר באמצעותה רק להתעלם אך לא לתפוס את הסיגנל SIGKILL ג.
- ד. הפונקציה מחזירה ערך המעיד על כישלון במקרה שמנסים לתפוס או להתעלם מהסינגל SIGKILL

שאלה 9

להלן אלגוריתם להקצאת שטחי זיכרון: השטח המוקצה הראשון יישמר בתחילת הזיכרון. בכל פעם שנרצה להקצות, נאתר את השטח הפנוי הרצוף הגדול ביותר בזיכרון, ונקצה שטח חדש החל מאמצע השטח הפנוי הרציף הגדול ביותר שמצאנו. אם קיימים מספר שטחים כאלה, בוחרים אחד באקראי. בחרו טענה נכונה לגבי הפרגמנטציה (fragmentation) שתיווצר בזיכרון.

- א. השיטה ״תפזר״ שטחים תפוסים על פני הזיכרון ולכן תיצור תמונת פרגמנטציה דומה לזו של אלגוריתם ה first fit.
- ב. השיטה ייתרכזיי שטחים תפוסים בתחילת הזיכרון ולכן תיצור תמונת הפרגמנטציה דומה לזו של אלגוריתם ה worst fit.
- ג. השיטה מתארת בעצם את אלגוריתם ה-next fit ולכן תמונת הפרגמנטציה תהיה לגמרי זהה לזו של שיטת ה-next fit.
- ד. השיטה "תפזר" שטחים תפוסים על פני הזיכרון ותגרום לאי-יכולת להקצות שטחים תפוסים מהר יותר משיטת ה- first fit.
- ה. השיטה "תרכז" שטחים תפוסים בתחילת הזיכרון ותגרום לאי-יכולת להקצות שטחים תפוסים מהר יותר משיטת ה-worst fit.

להלן הפרוטוקול לפתרון בעיית הקטע הקריטי:

Process 0	Process 1
while (1){	while (1){
flag [0] = 1;	flag [1[= 1;
turn = 0;	turn = 1;
while ((flag[0]==0) xor (flag[1]==turn))	while ((flag[1]==1) xor (flag[0]==turn))
; // wait	; // wait
<critical section=""></critical>	<critical section=""></critical>
flag[0] = 0;	flag[1] = 0;
}	}

כאשר המערך [2] flag והמשתנה turn משותפים לשני התהליכים ומאותחלים ל 0. בחרו את הטענה הנכונה:

- א. הפתרון מקיים את התנאי של המניעה ההדדית בלבד.
- ב. קיימים מקרים בהם הפתרון לא תמיד מקיים את התנאי של המניעה ההדדית.
 - ג. הפתרון הוא פתרון סביר.
 - . (deadlock) והן לקיפאון (starvation) ד. הפתרון עשוי לגרום הן להרעבה

אמת של xor הערה: לפניכם טבלת

xor	0	1
0	0	1
1	1	0

שאלה 11

נניח שבכל i-node יש 10 מצביעים ישירים, בנוסף למצביעים לא ישירים. גודל בלוק הוא b בתים ומספר המצביעים שניתן לשמור בבלוק הוא n. הביטוי המתמטי עבור גודל הקובץ המקסימאלי שניתן לשמור בעזרת ה i-node הוא:

$$b \times (10+10n+10n^2+10n^3)$$
 .x

$$10 \times (b+bn+bn^2+bn^3)$$
 .ם

$$b \times (10 + n + n^2 + n^3)$$
 .

$$10+n+n^2+n^3$$
 .7

לפניכם פרוטוקול שהוצע לפתרון בעיית הפילוסופים הסועדים:

```
#define N
                                  // number of philosophers
#define LEFT
                  (i-1) % N
                                   // number of it's left neighbor
#define RIGHT (i+1) % N
                                  // number of it's right neighbor
#define THINKING
                                  // philosopher is thinking
#define HUNGRY
                                   // philosopher is trying to get forks
                          1
#define EATING
                          2
                                  // philosopher is eating
typedef int semaphore;
                                  // semaphores are a special kind of int
                                 // array to keep track of everyone's state
int state[N];
semaphore mutex = 1;
                                 // mutual exclusion for critical regions
semaphore s[N];
                                 // one semaphore per philosopher
void philosopher (int i){
                                 // i: which philosopher (0 to N-1)
      while (TRUE){
                                 // repeat forever
        think();
                                 // philosopher is thinking
        take forks();
                                 // acquire two forks or block
                                 // yam-yam, spaghetti
        eat();
                                 // put both forks on table
        put_forks();
}
void take forks(int i){
                                 // i: which philosopher (0 to N-1)
                                 // enter critical region
     down(&mutex);
     state[i] = HUNGRY;
                                 // record fact that philosopher i is hungry
                                 // try to acquire 2 forks
     test(i);
                                // exit critical region
     up(&mutex);
                                // block if forks were not acquired
     down(&s[i]);
void put forks(int i){
                                // i: which philosopher (0 to N-1)
     down(&mutex);
                                // enter critical region
     state[i] = THINKING;
                                // philosopher has finished eating
     test(LEFT);
                                // see if left neighbor can now eat
     test(RIGHT);
                                // see if right neighbor can now eat
     up(&mutex);
                                // exit critical region
void test(int i){
                                // i: which philosopher (0 to N-1)
     if (state[i] == HUNGRY \&\&
        state[LEFT] != EATING &&
        state[RIGHT] != EATING){
         state[i] = EATING;
          up(\&s[i]);
```

.test(RIGHT) הועברה אחר הועברה אחרה state $[i]= ext{THINKING}$ השורה put_forks נניח כי בפונקציה מה חשינוי על הפתרון כאשר נתון כי ישנם 2 תהליכי פילוסופים בלבד?

- א. השינוי יתבטא בהורדה קלה ברמת המקביליות של הפתרון.
- ב. השינוי יתבטא בעלייה קלה ברמת המקביליות של הפתרון.
 - ג. אין לשינוי כל השפעה על הפתרון.
 - ד. השינוי עשוי להביא לקיפאון במקרה של 2 תהליכים.

שאלה 13

.bytes להלן טבלת הדפים של תהליך רץ. כל המספרים הם עשרוניים. כל הכתובות הן כתובות ב גודל דף הוא 1024bytes.

Virtual Page Number	Valid Bit	Reference Bit	Modify Bit	Page Frame Number
0	1	1	0	4
1	1	1	1	7
2	0	0	0	-
3	1	0	0	2
4	0	0	0	-
5	1	0	1	0

חשבו את הכתובות הפיסיות, אם זה אפשרי, של הכתובת הוירטואלית5499:

- א. 379
- ב. 173
- د. 28
- ד. 14

שאלה 14

shortest job) SJF מה תהיה התחזית לריצה הבאה אם האלגוריתם לתזמון תהליכים הוא גרסה של aging מה תהיה התחזית (first

- 36, 4, 4, 12 הריצות שהיו בפועל (משמאל לימין) הן בפועל
 - a=1/4 •
 - שפי זמן התחלתי הוא 36
 - א. 10.5
 - ב. 12
 - 13.5 .λ
 - 24 .7

: lstat בחרו טענה נכונה לגבי הפונקציה

- א. היא מחזירה אינפורמציה על קובץ. אם הקובץ הוא קישור סימבולי לקובץ אחר, היא מחזירה אינפורמציה על קובץ הקישור ולא על הקובץ שהקישור מצביע עליו.
- ב. היא מחזירה אינפורמציה על קובץ. אם הקובץ הוא קישור סימבולי לקובץ אחר, היא מחזירה אינפורמציה על הקובץ שהקישור מצביע עליו ולא על קובץ הקישור.
- ג. היא תומכת במספר בלתי מוגבל של קישורים סימבוליים בנתיב הצבעה לקובץ רגיל (לדוגמה, קישור סימבולי אשר מצביע לקישור סימבולי אשר מצביע לקובץ רגיל)

 רגיל)
 - ד. היא לא מחזירה ערך כישלון אף פעם

שאלה 16

חשבו את גודל הדף (page) במערכת עם ניהול זיכרון באמצעות חלוקה לקטעים בשיתוף עם דפדוף (segmentation with paging) כאשר:

- אורך מילת הזיכרון הוא 2 בתים (bytes).
- אורך הכתובת המדומה (virtual address) הוא 36 סיביות (bits).
- מספר סיביות של הכתובת המדומה המיועד להיסט (offset) הוא 10.
- .16 הוא segment descriptor מספר הסיביות של הכתובת המדומה המיועד ל
 - מספר הסיביות של הכתובת המדומה לציון הדף הוא 10 סיביות.
 - 4Kbytes א
 - 2Kbytes .z
 - د. 1Kbytes
 - 0.5Kbytes .ד

שאלה 17

תהליך במצב zombie במערכת הפעלה UNIX הינו:

- א. כל תהליך שאיננו מתקדם עקב מחסור במשאב מסוים
- ב. כל תהליך אשר תהליך האב שלו "מת" (סיים את עבודתו)
 - ג. כל תהליך שקיבל סיגנל SIGKILL ג.
- ד. כל תהליך שסיים את עבודתו אבל תהליך האב שלו עדיין רץ ולא "אסף" אותו (ביצע wait)

תהי S קבוצת תהליכים במערכת עם ניהול זיכרון באמצעות דפדוף (paging). לכל אחד מהתהליכים S בקבוצה זמן ריצה סופי. התהליכים הללו רצים במקביל בשתי מערכות כאשר המערכת הראשונה משתמשת באלגוריתם האופטימלי להחלפת הדפים (page replacement) ואילו המערכת השנייה משתמשת באלגוריתם S (least recently used) LRU משתמשת באלגוריתם

(launching) **הערה:** נניח כי התהליכים רצים במקביל בשתי המערכות. הכוונה היא שזמני השיגור $s(i) \in S$ של כל אחד מ $s(i) \in S$ זהים בשתי המערכות. בנוסף, שתי המערכות זהות לחלוטין למעט האלגוריתם להחלפת הדפים.

בהנחה ש S הם התהליכים היחידים במערכת, בחרו טענה נכונה:

- א. בכל זמן נתון מספר פסיקות הדף במערכת הראשונה יהיה קטן או שווה למספר פסיקות הדף במערכת השנייה.
- ב. מספר פסיקות הדף במערכת הראשונה יהיה קטן או שווה למספר פסיקות הדף במערכת השנייה עם סיום כל התהליכים של S בשתי המערכות (ולא בכל רגע נתון). זאת בתנאי שמדיניות החלפת הדפים במערכת היא גלובלית.
- ג. מספר פסיקות הדף במערכת הראשונה יהיה קטן שווה למספר פסיקות הדף במערכת השנייהעם סיום כל התהליכים של S בשתי המערכות (ולא בהכרח בכל רגע נתון). זאת בתנאי שמדיניות החלפת הדפים במערכת היא לוקאלית.
- ד. מספר פסיקות הדף במערכת הראשונה יהיה קטן שווה למספר פסיקות הדף במערכת השנייה
 עם סיום כל התהליכים של S בשתי המערכות (ולא בכל רגע נתון). וזאת בלי תלות אם המדיניות
 להחלפת הדפים היא לוקאלית או גלובאלית.

שאלה 19

כמה משטחים (platters), מסלולים (tracks) וגזרות (sectors) נדרשים כדי לאכסן (platters), מסלולים כמה משטחים (בתים, מסלולים מסלולים, כל אחת בת 200 בתים, אם כל גזרה מאכסנת B=512 בתים, כל משטח כולל S=96 גזרות, ויש S=96 משטחים. הניחו שרשומה אינה מתפצלת בין גזרות – היא חייבת להיות בגזרה אחת.

- א. 7 משטחים, 790 מסלולים, 75,788 גזרות
- ב. 8 משטחים, 790 מסלולים, 75,788 גזרות
- 275,788 משטחים, 790 מסלולים, 75,788 גזרות ::
- ד. 8 משטחים, 790 מסלולים, 75,500 גזרות

המשך הבחינה בעמוד הבא

בחרו את החיסרון המשמעותי ביותר ביישום קבצים באמצעות רשימה משורשרת הממומשת בעזרת (linked list allocation using an index) אינדקס

- איטית. (sequential access) איטית
- .(random access) ב. העדר אפשרות לבצע קריאה
 - ג. ריסוק חיצוני (external fragmentation) של הדיסק.
 - ד. צורך בהחזקת טבלה גדולה בזיכרון.

בהצלחה!