

304680473

מספר התלמיד הנבחן
רשום את כל תשע הספרות

האוניברסיטה
הפתוחה



ד' באדר תשע"ח

529 מוס' שאלון - 22

בפברואר 2018

87 מוס' מועד

סמסטר 2018א

20594 / 4

שאלון בחינת גמר

20594 - מערכות הפעלה

משך בחינה: 3 שעות

בשאלון זה 10 עמודים

מבנה הבחינה:

הראו בעיון לפני שתתחלו בפתרון הבחינה!

א. המבחן מורכב משלושה חלקים.

ב. בחלקים א ו- ב מופיעות שאלות פתוחות. ענו תשובות מלאות, בכתב קרייא ובקיצור נמץ. אין חובה להשתמש בכל השירותים המוקצרים לצורך התשובות, אך אין לחרוג מהמקום המוקצה.

ג. בחלק ג (שאלות אמריקאיות) عليיכם לבחור בכל פעם בתשובה יחידה מבין התשובות המוצעות ולהקייפ בעיגול אתאות התשובה שבחרתם.

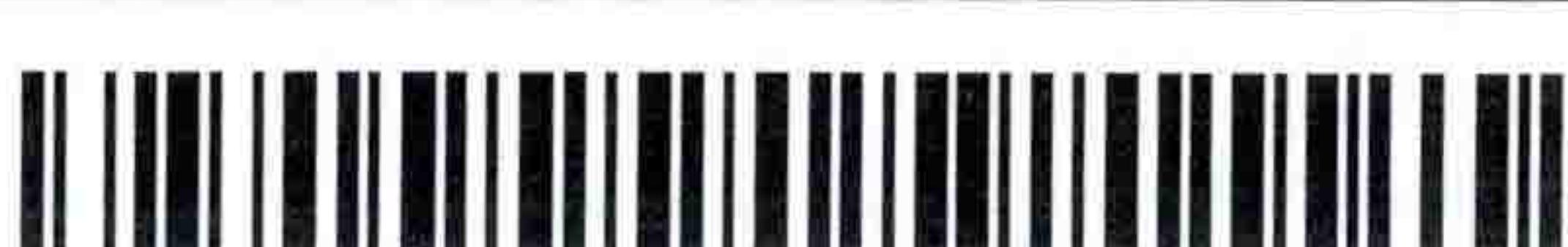
בצלחה !!!

חומר עזר:


כל חומר עזר אסור בשימוש, פרט למחשבון, שאינו אוצר מידע.

החזירו

למשגיח את השאלון
וכל עזר אחר שקיבלתם בתור מחברת התשובות



N101153242
ת.ז: 304680473
סידורי מוסף 4



חלק א (55 נקודות)

ענו על שלוש השאלות 1-3.

שאלה 1 (25 נקודות)

משמעות סמפור (semaphore) לティיאום בין מספר תהליכיוןים באותו תהליך ע"י שימוש ב-pipe. אפשר להגביל את השימוש שלכם על ידי כך שלא יתמודם במוניה של סמפור אשר ערכו גדול PIPE_BUF.

תזכורת:

Function: int **pipe** (*int filedes* [2])

The `pipe` function creates a pipe and puts the file descriptors for the reading and writing ends of the pipe (respectively) into `filedes` [0] and `filedes` [1].

If successful, `pipe` returns a value of 0. On failure, -1 is returned.

Reading or writing pipe data is *atomic* if the size of data written is not greater than `PIPE_BUF`. This means that the data transfer seems to be an instantaneous unit, in that nothing else in the system can observe a state in which it is partially complete. Atomic I/O may not begin right away (it may need to wait for buffer space or for data), but once it does begin it finishes immediately.

Reading or writing a larger amount of data may not be atomic; for example, output data from other processes sharing the descriptor may be interspersed. Also, once `PIPE_BUF` characters have been written, further writes will block until some characters are read.

לְאַבְנֵת אֶת־בָּנָה

```
typedef struct Semaphore_t {
    int ds[2];
} Semaphore;
```

ds[0] ←
ds[1] ←

```
int sem_init(Semaphore* sem, int count) { //assume pointer is allocated
```

```
    int fd = pipe(sem->ds);
```

```
    for (int i = 0; i < count; i++)
```

```
        write(sem->ds[1], "a", 1);
```

```
}
```

①

many sys calls
for write

```
int sem_wait(Semaphore* sem) {
```

```
    char c;
```

```
    read(sem->ds[0], &c, 1);
```

```
    return 1;
```

```
}
```

```
int sem_post(Semaphore* sem) {
```

```
    write(sem->ds[1], "a", 1);
```

```
    return 1;
```

```
}
```

הערה: אין צורך לכתוב קוד המטפל בכישלון של קריאות מערכת וגם אפשר להניח שהתוכנית שמשתמשת בסמפורים מתעלמת מ-SIGPIPE.

לְחַנּוּתָהָרָאֵבָה

שאלה 2 (20 נקודות)

(4 נק') א. מהו מספר שורות זיכרונות המטמון בהן נדרש לעורך חיפוש מקבילי במקרה של זיכרונות אסוציאטיביים באופן מלא (full associative)? נמקו.

כל שורה יתאפשר גישה לכול זיכרונות (fully associative) בודק כל זיכרון ובודק אם הוא מושך

(4 נק') ב. מהו מספר שורות זיכרונות המטמון בהן נדרש לעורך חיפוש מקבילי במקרה של זיכרונות אסוציאטיביים בעל מיפוי ישיר (direct mapping)? נמקו.

השורה יתאפשר גישה לזכרונות ייחודיים (direct mapping) על ידי הash function ✓

(4 נק') ג. תארו את אלגוריתם best-fit linked-list לניהול שטחי זיכרונות באמצעות רשימה משורשת.

אלגוריתם best-fit linked-list (linked-list) מבודד שטחים ומאפשר גישה לכולם. כל שטח כפוי לרשימה (linked-list) של שטחים אחרים. שטחים לא-רלוונטיים (non-relevant) לא יתאפשר גישה אליו. ✓

(4 נק') ד. איזה סוג של ריסוק (fragmentation) מייצר אלגוריתם best-fit לניהול שטחי זיכרונות באמצעות רשימה משורשת ומהי השפעתו על מפת השטחים הפנויים/התפוסיים?

אלגוריתם best-fit יתאפשר גישה לכולם (best fit) אך יתאפשר גישה לחלקם בלבד (compaction). שטחים לא-רלוונטיים (non-relevant) יתאפשר גישה אליו (external). ✓

(4 נק') ה. דרגות האלגוריתמים לפינוי דפים לפי ביצועים במובן של יכולת חיזוי בסדר יורד: WSClock, Least Recently Used, FIFO, אופטימלי.

best	lowest
	WSClock
	Least Recently Used
worst	FIFO

הערה: יכולת חיזוי טובה של אלגוריתם להחלפת דפים פירושו כמות קטנה של פסיקות דף (page faults).

EXCERPTS FROM

שאלה 3 (10 נקודות)

הוכיחו או הפריכו את הטענות הבאות לגבי אלגוריתם הבנקאי (banker algorithm) :

- (5 נק') א. אם אלגוריתם הבנקאי גילה שהמערכת נמצאת במצב לא בטוח, אז בהכרח שהמערכת תגיע לקייפאון בעתיד.

- (5 נק') ב. אם אלגוריתם הבנקאי גילה שהמערכת נמצאת במצב בטוח, אנחנו יודעים בוודאות

שנוכן לסייע את הריצה בלי להגיע ל-deadlock.

המשך הבדיקה בעמוד הבא

לְאַבְנֵת כָּסִיר טַלְיָה

חלק ב (25 נקודות)

ענו על חמיש השאלות הבאות. משקל כל שאלה 5 נקודות.

שאלה 4

הסבירו מה עושה התוכנית ורשמו מה יהיה הפלט שלה:

```
#include <unistd.h>
#include <fcntl.h>
#include <sys/types.h>

int main()
{
    int file=0;
    if((file=open("testfile.txt",O_RDONLY)) < -1)
        return 1;

    char buffer[19];
    if(read(file,buffer,19) != 19) return 1;
    printf("%s\n",buffer);

    if(lseek(file,10,SEEK_SET) < 0) return 1;

    if(read(file,buffer,19) != 19) return 1;
    printf("%s\n",buffer);

    return 0;
}
```

כasher

```
$ cat testfile.txt
This is a test file that will be used
to demonstrate the use of lseek.
```

529
אלין
87.31.16

buffer
This is a test file :/
test file that will

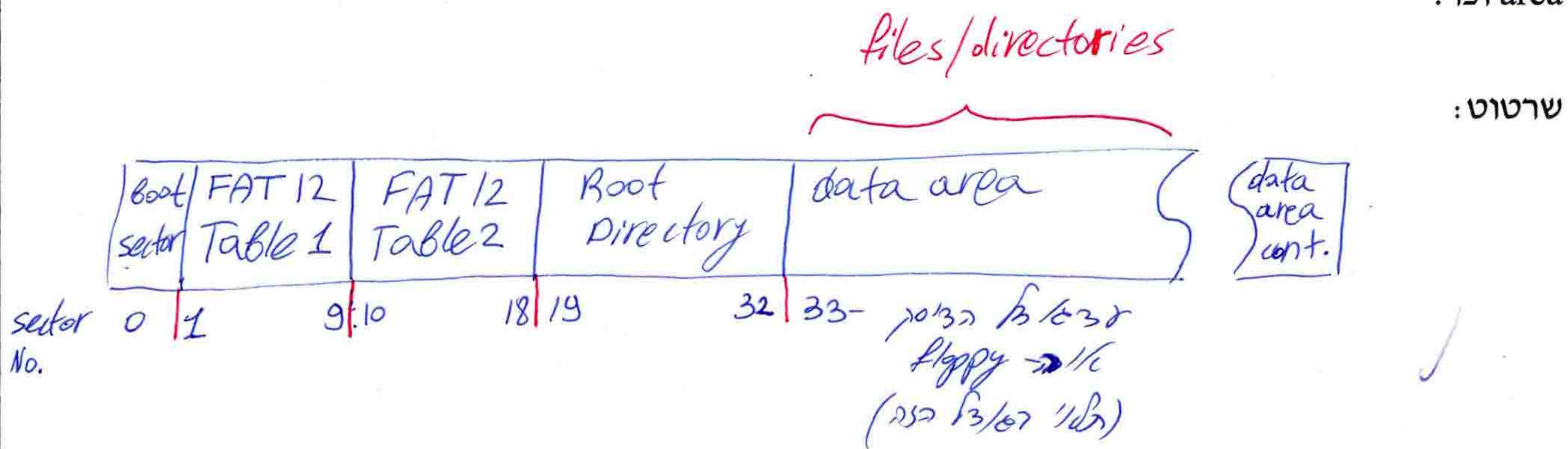
תְּאַלֵּה לְכָהִיר אֲשֶׁר

שאלה 5

מערכת הפעלה לינוקס מוחזיקה טבלת תהליכיים (process table) בזיכרון הראשי. כל כניסה (entry) בטבלה תהליכיים מכילה מידע על סיגנלים אשר נשלחו לתהליך. מדובר מידע זה נכלל בטבלת התהליכיים וכתווצה מכך נמצא כל הזמן בזיכרון הראשי הסבירו איזו עיטה הייתה מתוערת היה והינו שומרם מידע זה יחד עם דפים וריגלים של תהליכיים.

שאלה 6

שרטו את ה-layout של מערכת הקבצים FAT12, כולל boot sector, FAT12, TABLE, ROOT DIRECTORY, DATA AREA וכו'.



לען כהן סאליבן

שאלה 7

האם יתכן מצב קיפאון אם אף תהליך במערכת אינו רשאי להחזיק במשאב כלשהו בעודו מבקש משאב חדש? הסבירו את תשובהכם.

מצב קיפאון (Hold & wait) מצב קבוק (deadlock)
מצב קבוק (deadlock) מצב קבוק (deadlock)

שאלה 8

מערכת הקבצים של מערכת הפעלה מסויימת משתמשת בשיטת ה I-node.

$2^8 = \text{addresses per block}$
 (256)

$\left\{ \begin{array}{l} \text{block size} = 2^1 \text{ byte} \\ \text{address size} = 2^3 \text{ bytes} \end{array} \right.$

- גודל הבלוק במערכת הקבצים הוא 2 Kbyte

- כתובות הבלוק היא 8 בתים (bytes)

- 10 שדות של ה I-node יכולים להחזיק ישירות כתובות הבלוק בדיסק.

- שלושה שדות נוספים:

- שדה הנועד להחזיק את הכתובת של ה-block

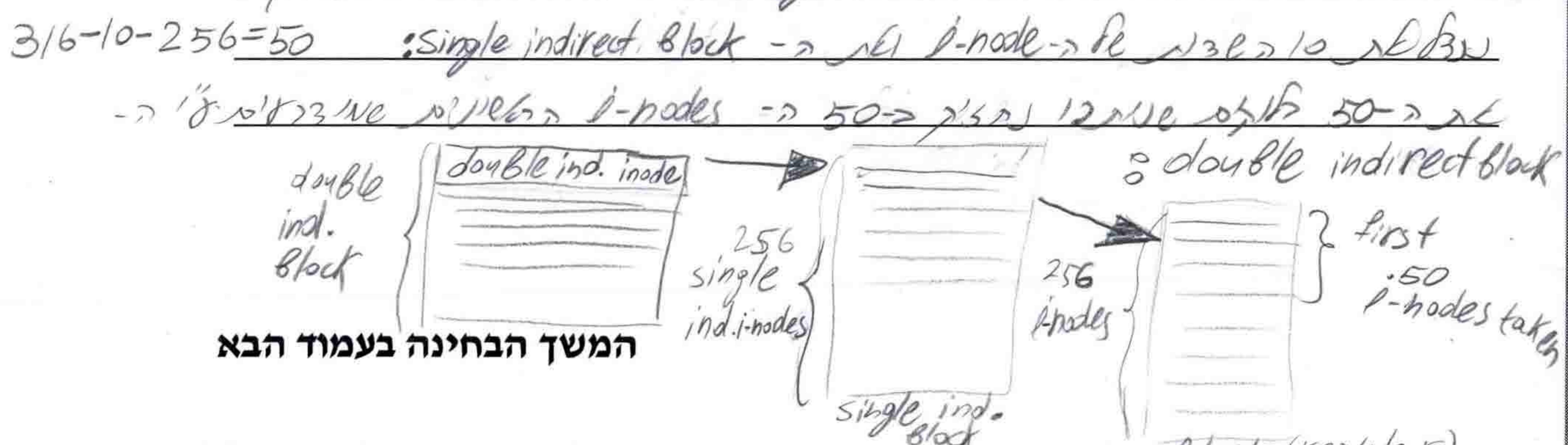
- שדה הנועד להחזיק את הכתובת של ה-block

- שדה הנועד להחזיק את הכתובת של ה-block

גודלו של קובץ מסוים במערכת הוא 632 Kbyte. חשבו מהי כמות הבלוקים שדרישה להחזקת קובץ זה במערכת הקבצים ותנו הסבר מפורט לחישוב.

$$\text{amount of addresses per block} = \frac{2 \text{ Kbyte}}{8 \text{ bytes}} = 2^8 = 256 \text{ addresses per block}$$

$$\text{Required blocks count} = \frac{632 \text{ Kbytes}}{2 \text{ Kbytes}} = 316 \text{ blocks}$$



Total blocks = $316 + 2 + 1 = 319$ blocks

single double ind. ind.



לְאַמְתִּיבָה

חלק ג (20 נקודות)

ענו על ארבע שאלות רב-ברירה (אמריקאיות). משקל כל שאלה 5 נקודות.
בכל שאלה יש לבחור את התשובה הנכונה ולהקיף בעיגול את אות התשובה שבחרתם.

שאלה 9

בחרו תשובה נכונה לגבי סיום תהליכיים במערכת הפעלה Linux:

- waitall*
א. תהליך אב ימתין עד לסיום של אחד מבניו (בן כלשהו) אם קרא ל (...,-1) *waitpid*
- wait for process group*
ב. תהליך אב ימתין עד לסיום של אחד מבניו (בן כלשהו) אם קרא ל (...,-0) *waitpid*
- pid=1*
ג. תהליך אב ימתין עד לסיום של תהליך בן כלשהו אם קרא ל (...,-1) *waitpid* *או הפוך*
- ד. כל התשובות הקודמות הן נכונות



שאלה 10

נתונים שני תהליכיים שרצים במקביל. להלן הפסאודו-קוד שלהם:

Process 0	Process 1
<pre>while (1){ for (i=0; i<N; i++) down(Si); /* Critical section */ for (i=N-1; i>=0; i--) up(Si); }</pre>	<pre>while (1){ for (i=0; i<N; i++) down(Si); /* Critical section */ for (i=N-1; i>=0; i--) up(Si); }</pre>

כאשר Si –ים הם N סמלרים בינאריים שאוטחלו ל 1. N הוא מספר טבעי גדול מ-2.

בחרו את הטענה הנכונה:

- א. שני תהליכיים יכולים לשחות בו-זמנית בקטע קריטי.
- ב. שני תהליכיים עלולים להיכנס למצב קיפאון.
- ג. הפרוטוקול מבטיח קידימות של Process 0 על פני התהליך המتأخرה.
- ד. הפרוטוקול מבטיח קידימות של Process 1 על פני התהליך המتأخرה.

ה. הפרוטוקול פוטר את בעיית הקטע הקריטי.

אך לא מוגדר אייזן מסע ו-N סמלרים

מִשְׁמָרַת אֶתְנָהָרָה

שאלה 11

בחרו טענה נכונה לגבי מערכות ה-Exokernel :
א. מערכות ה-Exokernel מספקות לרמת אפליקציה את משאבי המערכת zusätzlich לאוסף מצומצם מאוד של אבstractionיות. כתוצאה לכך לאפליקציות ניתן מרחב תמרון גדול מאוד בכל הנוגע לניהול משאבי המערכת.

ב. גרעין מערכת הפעלה מתפקיד כדוור המרים בקשות של לkers (תהליכיים) אל שרתים עצמאיים המפוזרים במערכת, כגון שרתים ניהול הקבצים, ניהול הזיכרון וכו'. האבstractionיות ממומשות ע"י שרתים בלבד.

ג. מערכות ה-Exokernel מתאפיינות באספקה של שירותים של המכונה המורחבת ושל המכונה המדומה.

ד. כל התשובות הקודמות הן נכונות.

שאלה 12

מהי הסיבה העיקרית לשימוש ב-DMA?

א. אפשר למעבד לróż בקצב מהיר יותר.

ב. שיפור ביצועי המערכת ע"י הגדלת המקבילות.

ג. הקטנת העומס על הזיכרון הראשי.

ד. מיקסום הניצול של שטח הדיסק.

בצלחה !

לְאַמְתִּיבָה וְלְאַמְתִּיבָה