תאריך הבחינה: 05/07/2019. שם המרצה: רוברט יעקובשוילי. שם הקורס: מערכות הפעלה. מספר הקורס: 37111631. שנה: 2019 סמסטר: ב' מועד: א' משך הבחינה: שלוש שעות. חומר עזר: אסור.

במבחן 6 שאלות. יש לענות על כולן. הניקוד של כל סעיף מופיע בסוגריים, לצד הסעיף.

.1 חוטים, מרוץ תהליכים, תזמונים (18 נקודות)

במערכת הפעלה עם CPU יחיד, 2 חוטים מריצים במקביל פונקציית incrementCounter כמתוארת לפניך. הפעלה עם CPU מסוג int הפונקציה עובדת עם משתנה גלובלי counter מסוג int

```
\label{eq:counter} \begin{split} & \text{int incrementCounter(void * arg) } \{\\ & \text{(void)arg;} \\ & \text{for (int } j = 0; j < 100; j++) \\ & \text{counter++;} \\ & \text{return } 0; \end{split}
```

- א. (5 נקי) מהו הערך המקסימלי שעשוי להתקבל בתום הרצת הקוד? תארי תרחיש מתאים.
- ב. (5 נקי) מהו הערך המינימלי שעשוי להתקבל בתום הרצת הקוד? תארי תרחיש מתאים.
- ג. (2 נקי) צייני שתי שיטות סנכרון סטנדרטיות שניתן להשתמש בהן כדי לסנכרן את הגישה למשתנה: counter
 - ג. (3 נקי) מהי השיטה היעילה ביותר לסנכרון במקרה שלנו?
- להכיל counter ויגרום רמכי race-conditions שימנע את (scheduling) ויגרום להכיל אלגוריתם תזמון אלגוריתם מנכרון. את התוצאה הנכונה גם ללא שימוש במנגנון סינכרון.

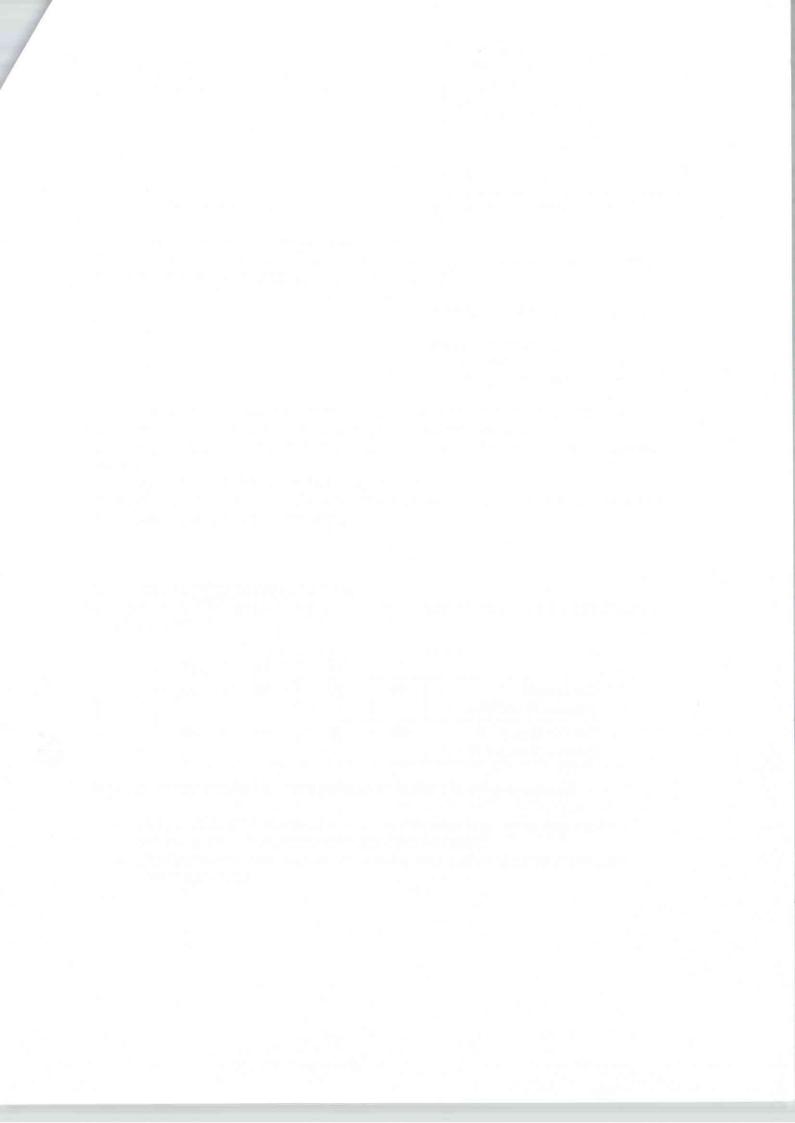
2. תזמונים <u>זמן אמת (RT) (RT) נקודות)</u> מערכת הפעלה RT-OS רצה ב-CPU יחיד ומריצה

מערכת הפעלה RT-OS רצה ב-CPU יחיד ומריצה את המשימות T1 (העדיפות הגבוהה ביותר) עד T4 (העדיפות הנמוכה ביותר).

	T1	T2	Т3	T4
Job Length, Abstract Time Units.	10	20	20	40
Job Deadline, Abstract Time Units.	100	80	60	50

.deadlines ביותר למערכת ההפעלה הזו הוא לסיים את כל המשימות עד לסיום ה

- א. (3 נק') כל המשימות מתחילות באותו הזמן ומערכת ההפעלה הזו מריצה אותן לפי סדר העדיפויות. איזה משימות יספיקו לסיים בזמן? פרט את החישוב.
- ב. (9 נק') הצע אלגוריתם תזמון שפותר את הבעיה שמצאת בסעיף א' וסרטט לו גרף ביצוע משימות כתלות בזמן.



3. ניהול זכרון (19 נקודות)

א. (9 נקי) מערכת ההפעלה הקצתה לתהליך 3 דפי זכרון פיזיים. נתונה סדרת הגישות (משמול לימין): page faults - תאר את תוכן הזיכרון בכל שלב וחשב את מספר ה- 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 בעת השימוש באלגוריתמים:

- FIFO (נק') .i
- Optimal (נקי') .ii
- Last Recently Used ('בק') .iii
- ב. סמן נכון/לא נכון לגבי הטענות הבאות:

הגדלה של גודל דף הזכרון:

- pagetable מקטינה את גודל ה מקטינה (2 בק') .i
- page fault מורידה את זמן הטיפול ב-ii
- iii. (2 בק') מגדילה את הפרגמנטציה הפנימית של הדפים.
 - ג. סמן נכון/לא נכון לגבי הטענות הבאות ונמק בקצרה.
- Inverted Page-Tables במערכות עם TLB (בקי) i TLB (בקי). i
- MMU ב flush TLB בין החוטים של אותו התהליך לא דורש context switching (בק') .ii

4. סינכרוניזציה א' (16 נקודות)

אלגוריתם המאפייה (Lamport's Bakery algorithm) מיועד לפתור בעיית סנכרון, תוך שמירה על (mutual exclusion). הדרה הדדית (matual exclusion) וללא חשש לקיפאון (מטה מופיע מימוש נאיבי של האלגוריתם.

```
code of process i, i \in \{1,...,n\}

number[i] := 1 + \max \{number[j] \mid (1 \le j \le n)\}

for j := 1 to n \iff i \in \{1,...,n\}

for j := 1 to n \iff i \in \{1,...,n\}

await (number[j] = 0) \lor (number[j] > number[i])

\{i \in \{1,...,n\}\}

critical section

number[i] := 0
```

- א. (4 נק') פרט תרחיש שעלול ליצור קיפאון.
- ב. (4 בקי) פרט תרחיש שעלול ליצור הפרה של תנאי ההדרה ההדדית.
 - ג. (4 נק') תקן את הקוד כך שיפתור את שתי הבעיות.
- ד. (4 נק') האם האלגוריתם המתוקן הוא חופשי מהרעבה? אם כן, הוכח. אם לא, תן תרחיש שבו מתקיימת הרעבה.

5. סינכרוניזציה ב' (20 נקודות)

תוכורת: סמפור בינארי לא מתאים לשרת כ mutex בגלל שתי בעיות:

-חוט אחר יכול לשחרר את המנעול.

-חוט שמשתמש ברקורסיה נועל את עצמו בקריאה הרקורסיבית הבאה, ויוצר קיפאון (deadlock). mutex פותר את שתי הבעיות הנ"ל.

א. (15 נק') בנה באמצעות סמפור בינארי ומשתנים נוספים mutex שפותר את הבעיות הנ"ל. בפרט, על ה-mutex לעמוד בדרישות הבאות:

- -עובד נכון בפונקציות רקורסיביות.
- -יכול להשתחרר רק על ידי החוט שנעל אותו.
- -אם הינו נעול על ידי חוט מסויים, כל חוט אחר יצטרך להמתין בסמפור.
 - -עובד בסביבה של CPU יחיד.

:pseudo-code: או בשפת הבאות הבאות בצע את המשימות הבאות

- struct ReentrantMutex 3)
- initReentantMutex (ReentrantMutex* M) כתוב פונקציה: (2 בק')
- lockReentantMutex (ReentrantMutex* M) כתוב פונקציה: (5 בק')
- unlockReentantMutex (ReentrantMutex* M) כתוב פונקציה: (5 בק') כתוב

thread_id שלו הטיפוס . pthread_self()- שלו הרצאעות שלו thread_id שלו מוצא את היט מוצא את היט שלו הרא thread_id שלו הוא .pthread_t הוא

ב. (5 נק') השתמשו ב-CAS כדי לשדרג את CAS כדי לשדרג את multi CPUs בתרון שלכם לסביבת unlockReentantMutex

Compare-and-swap(w, old, new)

do atomically creating full read-write memory barrier

prev:=w
if prev = old
w:=new
return true
else

return false

6. שאלת קבצים (15 נקודות)

נתונה מערכת דמוית Unix שמשתמשת ב-I-node לארגון קבצים.

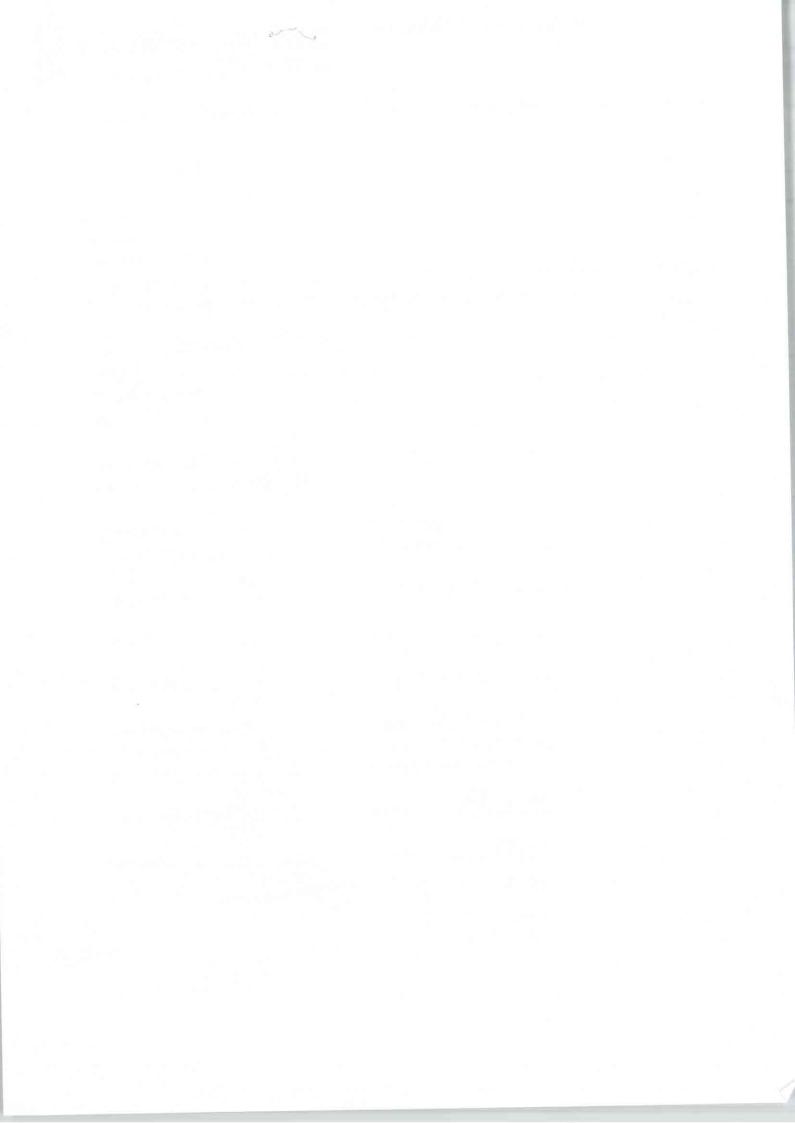
- .1K bytes גודל הבלוק הוא
- .4bytes גודל מספר הבלוק על הדיסק הוא
- של הקובץ. attributes של כל קובץ מאוחסן בבלוק אחד, שבתחילתו מאוחסנים ה-attributes של הקובץ.
 - ס לאחר מכן יש 10 מצביעים ישירות על בלוקים שמכילים נתונים;
 - ;single indirect pointers אחריהם, 16 מצביעים המשמשים כ
 - ;double indirect pointers אחריהם, 4 מצביעים שמשמשים כ-
 - .triple indirect pointers כי ולבסוף, 4 מצביעים המשמשים כ-

בסעיפים שלמטה יש לפרט את אופן החישוב, אך אין צורך להגיע לתשובה סופית - מספיק להגיע לביטוי מתאים, למשל: 1024*127*17*17*18*1024.

?את? מהו הגודל המקסימלי של הקובץ בבתים במערכת הזאת?

הפקולטה למדעי ההנדסה

: תאריך	
	ל: מזכירות הפקולטה למדעי ההנדסה
	(את:
	אלום רב,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	זנדון: דו"ח סיכום בחינה
ציון הסופי, אם כלל המרכיבים זמינים. אחרת, יוצגו על ציון אחר קבלת תוצאות בחינת מועד בי.	ב בתנונותנונהות מטה יוצגו על ה
D	
	שם הקורס: <u>אלככוג הפלט</u>
	מספר קורס:
	קורס :(חובה בחירה
	שם המרצה:
השתתפו בו סטודנטים	הקורס ניתן בסמסטר : אי בי קיץ ו סטטיסטיקות על ציון : סופי בחינה
בחינת מועד ב׳ מספר הנבחנים :	בחינת מועד א׳
בוטפו וונבווניט .	מספר נבחנים:
הציון הממוצע:	
סטיית תקן:	סטיית תקן:
אחוז הנכשלים:	אחוז הנכשלים:
האם ניתן פקטור כן/לא במידה וניתן על איזה מרכיב במידה וניתן מה משקלו	האם ניתן פקטור כן עלא במידה וניתן על איזה מרכיב
תאריך: _ 9, 7, 19	במידה וניתן מה משקלו חתימת המרצה:
תאריד <u>הלהוו</u> שורת 9, 07, 14	חתימת רמ"ח: <u>פרופי או אביו</u> בחלת המחלקה
שורת 19. 07. 19	רמצח המקלקוי להנדסת מערכות עק
WITT -	



1007 100 100 100 100 100 100 100 100 100
--

כנגב

