**שאלה 1 (5%)**

מה הוא התפקיד העיקרי של system calls במערכת מחשב?



ביצוע פעולות שדורשות הרבה מאוד זמן, למשל, כפל מטריצות ענקיות



ביצוע פעולות ב kernel mode בעקבות בקשת תהליכי משתמש



ביצוע פעולות ב user mode בעקבות בקשת תהליכי משתמש



ביצוע פעולות של מנהל המערכת



ביצוע פעולות הקומפיילר בזמן ביצוע קומפילציה

**משוב לבחירה בתשובה זו**

התשובה הנכונה:

ביצוע פעולות ב kernel mode בעקבות בקשת תהליכי משתמש

**שאלה 2**

תקין

5.00 נקודות מתוך 5.00

סימון שאלה

**תוכן השאלה**

**שאלה 2 (5%)**

מה היא הדרך הנכונה **והיעילה**למניעת deadlock במערכת עם 4 RESOURSES?



כל ה processes יוכלו לבקש ולהחזיר רק RESOURSE אחד בכל נקודת זמן של חיי התהליך



כל process יוכל לבקש לא יותר מ 3 RESOURCES



כל ה processes יוכלו לקבל את RESOURCES רק לפי סדר עולה ולשחרר לפי סדר יורד



כל process יוכל לבקש לא יותר מ 2 RESOURCES



כל process ישחרר את כל RESOURCES שהוא **החזיק בהם**(כולל אמצעי סנכרון) לפני שיסתיים

**משוב לבחירה בתשובה זו**

התשובה הנכונה:

כל ה processes יוכלו לקבל את RESOURCES רק לפי סדר עולה ולשחרר לפי סדר יורד

**מידע**

סימון שאלה

**טקסט מידע**

**שאלות 3-5**

בשביל כל אחת מ 3 בעיות ה synchronization  שבהמשך (שאלות 3,4,5), בחרו את אמצעי ה synchronization הטוב (היעיל) ביותר בכדי לפתור את הבעיה.

אין צורך לממש אלגוריתם שלם.

באלגוריתם יכולים להיות מבני נתונים/משתנים שהכרחיים לפתרון, אין צורך להתייחס אליהם.

אין צורך לדאוג כיצד מתאפשרת הגישה לאמצעי ה synchronization, צריך להניח שיש גישה.

**שאלה 3**

שגוי

0.00 נקודות מתוך 5.00

סימון שאלה

**תוכן השאלה**

**שאלה 3 (5%)**

צוות מתכנתים מפתח מערכת דיגיטלית לקביעת תורים לרופא. המערכת אמורה לאפשר לכל המעוניינים ביצוע מקבילי של פעולות הבאות:

1) קביעת תור לשעה פנויה.

2) ביטול תור.

3) הוספת תורים ע"י רופא. רופא לא מבטל תורים וכשמוסיף, תמיד מוסיף מספר משמעותי של תורים.

כל מי שמתחבר למערכת מטופל כ THREAD נפרד ואם אין מקום פנוי, הלקוח מתנתק ו- THREAD שנוצר עבורו במערכת מחכה לתור שיתפנה ויודיע לו דרך שליחת מסרון טלפוני.

הציעי אמצעי synchronization שיאפשר פתרון יעיל לדרישות



Lock and Condition variable with Condition broadcast



3 MUTEX (LOCK)



1 MUTEX (LOCK)



2 MUTEX (LOCK)



Lock and Condition variable with Condition signal

**משוב לבחירה בתשובה זו**

התשובה הנכונה: Lock and Condition variable with Condition broadcast

**שאלה 4**

הושלם

לא ניתן ציון

סימון שאלה

**תוכן השאלה**

**שאלה 3 - נימוק**

נמקי את בחירתך

**בשביל לקבוע תור לשעה פנויה צריך מנעול וכן שלא תתפס השעה הפנויה שניה לפני זה צריך גם כן מנעול  
וכן שהרופא מכניס ביחד זה לא משנה כי עדין מכניס לכל שעה פנויה ורק אם יש תורים  
לבטל תור לא צריך מנעול כי אפשר לבטל כמה שרוצים**

**שאלה 5**

תקין

5.00 נקודות מתוך 5.00

סימון שאלה

**תוכן השאלה**

**שאלה 4 (5%)**

במשחק מחשב קיימת בעיית מעבר בגשר צר שמעבר בו אפשרי רק בכיוון אחד.

מכוניות נעות במהלך משחק מחשב כשכל אחת היא THREAD נפרד. כשקיימות מכוניות הנוסעות בכיוון מסוים, מכוניות שרוצות לנוע בכיוון נגדי צריכות להמתין עד שייפסקו תנועת כל המכוניות בכיוון הנוכחי. ידוע מניסיון, שאף פעם לא מצטברות יותר מ 20 מכוניות שמחכות (אין צורך לטפל במקרה שמספרם יותר גדול), אבל מספרם יכול להיות גם נמוך יותר. כמו כן, אין צורך לדאוג למניעת התנגשויות בין המכוניות הנעות באותו כיוון.

בעזרת אילו אמצעי  synchronization ניתן לפתור בקלות  את הבעיה?

האמצעי/ם חייב/ים להיות יעילים ביותר.

עקרונית **מותר להשתמש בפתרון בנוסף גם במשתנים רגילים** ואין צורך להתייחס לזה.



2 Binary semaphore



2 MUTEX (LOCK)



Lock and Condition variable with Condition signal



Lock and Condition variable with Condition broadcast



2 Counting semaphore

**משוב לבחירה בתשובה זו**

התשובה הנכונה:

Lock and Condition variable with Condition broadcast

**שאלה 6**

הושלם

לא ניתן ציון

סימון שאלה

**תוכן השאלה**

**שאלה 4 - נימוק**

נמקי את בחירתך

כל צד מחכה לצד השני התנאי שאין מכוניות בצד הנגדי ואז מעוררים את כל המכוניות שבאותו צד   
(בחרתי לא להתיחס שזה עד 20 מכוניות)  
אפשרי גם  2 סמפורים בינארי שבזה נחסום את הכביש וכל צד יחכה לצד השני

**שאלה 7**

תקין

5.00 נקודות מתוך 5.00

סימון שאלה

**תוכן השאלה**

**שאלה 5 (5%)**

צוות מתכנתים מפתח מערכת ניהול מקומות חניה במבנה רב קומתי. המערכת אמורה לאפשר לכל המעוניין ביצוע מקבילי של פעולות הבאות מכמה עמדות שמחוברות לאותה מערכת:

1) תפיסת מקום חניה.

2) רישום לחניה בקומה מסוימת. בכל קומה קיימים כמה מקומות חניה.

3) קבלת הודעה טלפונית מתי שמקום חניה מתפנה.

4) שחרור מקום ביציאה מן החניון, גם לזה קיימות כמה עמדות שגם כן מחוברות לאותה מערכת.

כל המתחבר למערכת מטופל כ THREAD נפרד שמטפל בהרשמה/ביטול/שחרור מקום, מחכה עד שמקום מתפנה ושולח הודעה טלפונית . עקרונית **מותר להשתמש בפתרון בנוסף גם במשתנים רגילים** ואין צורך להתייחס לזה.

הציעי אמצעי synchronization שיאפשר פתרון יעיל לדרישות?



Lock and Condition variable with Condition signal



3 Counting semaphores



1 MUTEX (LOCK)



2 Counting semaphores



Lock and Condition variable with Condition broadcast

**משוב לבחירה בתשובה זו**

התשובה הנכונה: Lock and Condition variable with Condition signal

**שאלה 8**

הושלם

לא ניתן ציון

סימון שאלה

**תוכן השאלה**

**שאלה 5 - נימוק**

נמקי את בחירתך

התנאי שברגע שמתפנה מקום חניה משחרר את המנעול לאחד שביקש  
ושוב לבטל אין בעיה אפשר כמה שרוצים  בתפיסת מקום כולל בחירת קומה אין בעיה כי אם זה לא פנוי נכנס להמתנה עד שיתפנה

**שאלה 9**

תקין

5.00 נקודות מתוך 5.00

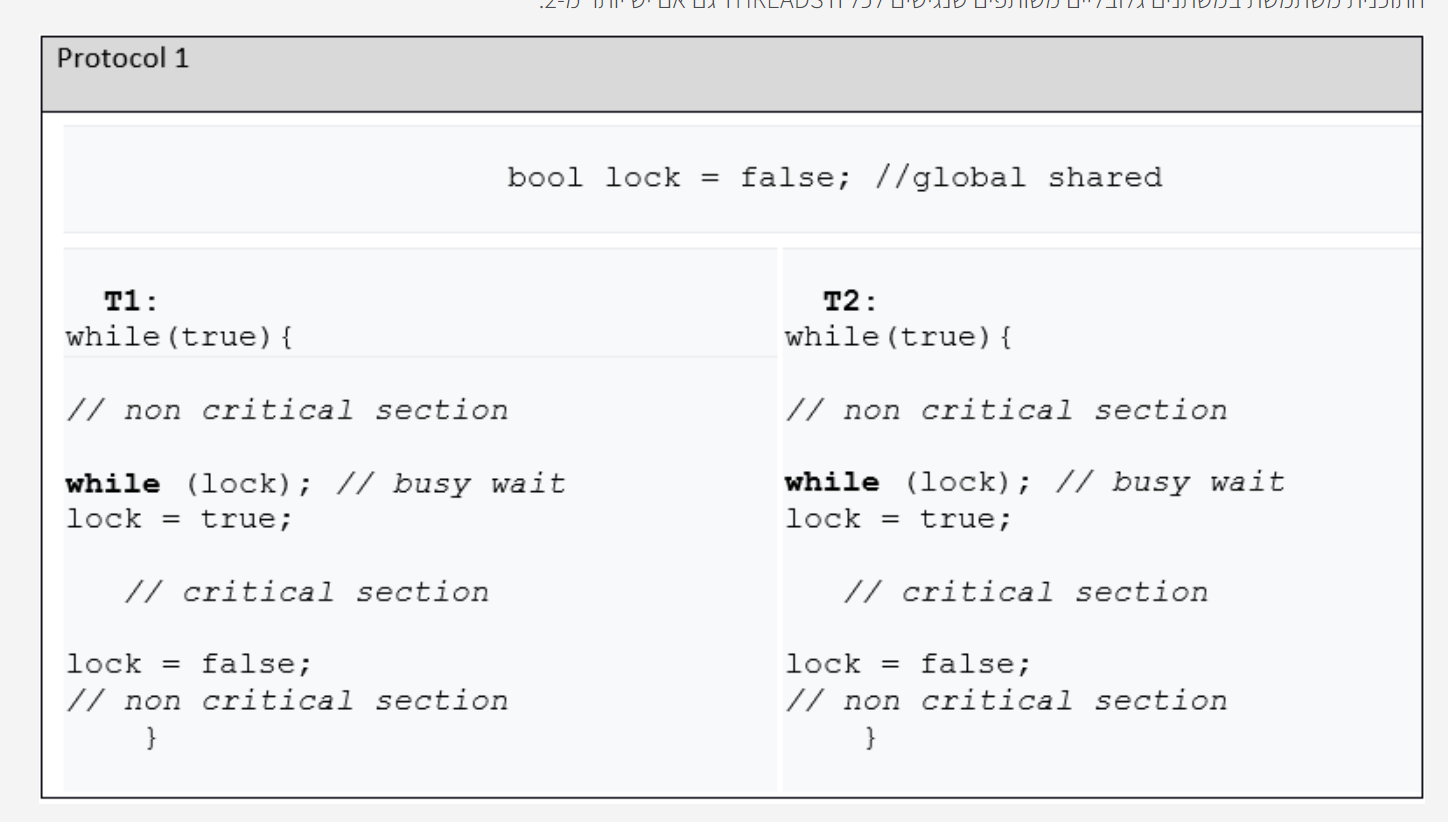
סימון שאלה

**תוכן השאלה**

**שאלה 6 (5%)**

נתונה התוכנית הבאה (pseudo code) שמיועדת ל SYNCHRONIZATION בין מספר THREADS (T1 ו T2 לדוגמא, **אבל יכולים להיות יותר**) שרצים במקביל. כל THREAD מבצע אותו פרוטוקול כניסה ויציאה ל  CRITICAL SECTION.

התוכנית משתמשת במשתנים גלובליים משותפים שנגישים לכל ה THREADS גם אם יש יותר מ-2.



בחרו את הטענה הנכונה:



הפרוטוקול פותר את בעיית ה CRITICAL SECTION



מספר threads יכולים לשהות בו-זמנית ב CRITICAL SECTION



שני THREADS לכל היותר יכולים לשהות בו-זמנית ב CRITICAL SECTION



הפרוטוקול פותר את בעיית ה CRITICAL SECTION אבל רק אם יש 2 THREADS ולא יותר



כמה threads עלולים להיכנס למצב DEADLOCK

**משוב לבחירה בתשובה זו**

התשובה הנכונה:

מספר threads יכולים לשהות בו-זמנית ב CRITICAL SECTION

**שאלה 10**

הושלם

לא ניתן ציון

סימון שאלה

**תוכן השאלה**

**שאלה 6 - נימוק**

נמקי את בחירתך

**קוד t1 יכנס ללולאה השניה ואת lock=false עדין לא יבצע  את lock=true לבנתיים t2 יכנס ויעבור את לולאת ההמתנה כי עדין  lock=false וכעת שניהם בקטע הקריטי באותו מידה יכולים להכנס עוד הרבה תהליכים שיתחילו את הקוד במקביל**

**שאלה 11**

תקין

10.00 נקודות מתוך 10.00

סימון שאלה

**תוכן השאלה**

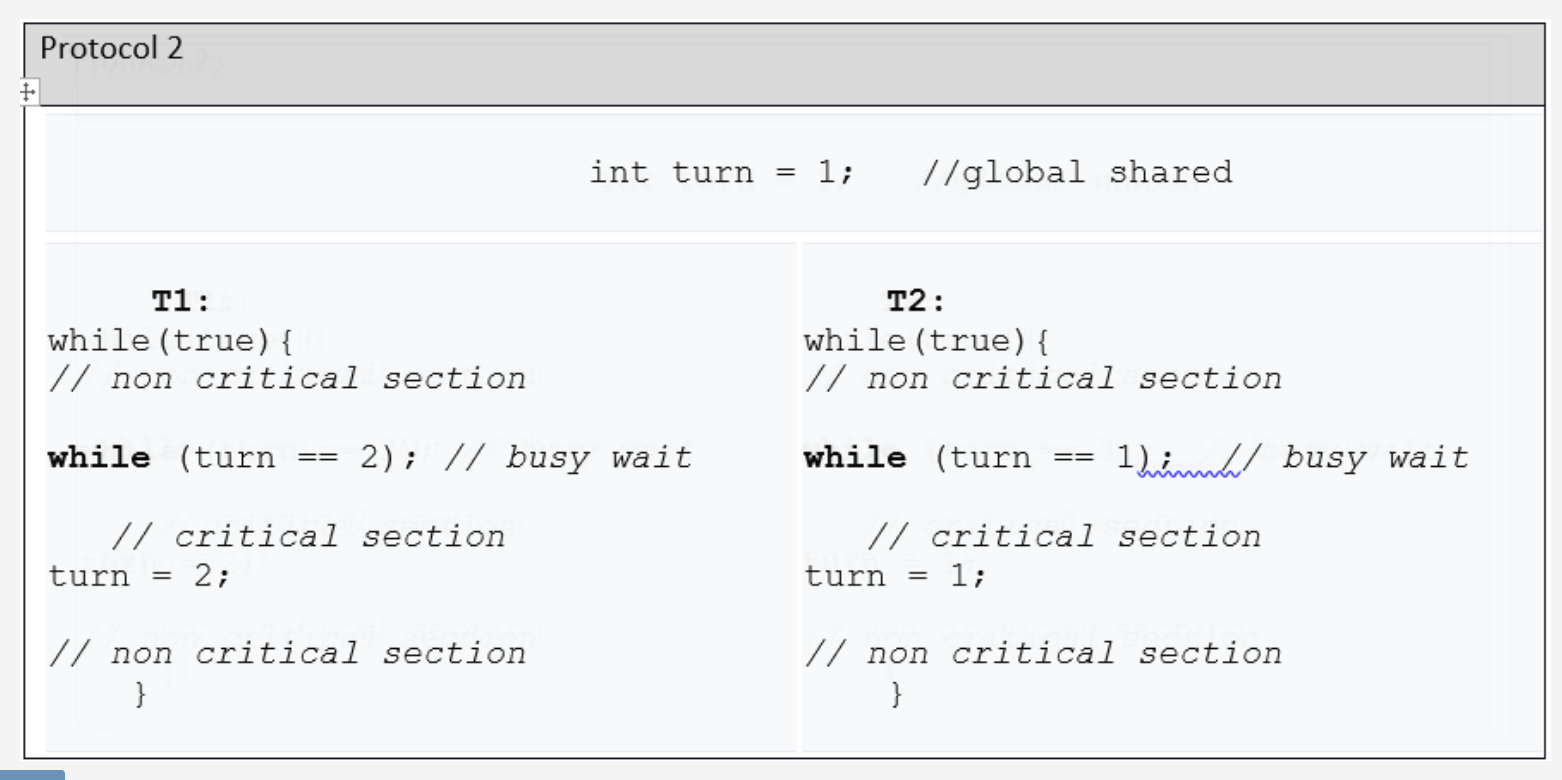
**שאלה 7 (10%)**

נתונה התוכנית הבאה (pseudo code) שמיועדת ל  SYNCHRONIZATION בין **שני** THREADS - T1 and T2 שרצים כל הזמן במקביל.

כל THREAD מבצע אותו פרוטוקול כניסה ויציאה ל  CRITICAL SECTION.

התוכנית משתמשת במשתנים גלובליים משותפים שנגישים רק ל 2 ה THREADS

בחרו את הטענה הנכונה:



הפרוטוקול מבטיח קדימות של T1 על פני המתחרה בכניסה ל CRITICAL SECTION



הפרוטוקול פותר את בעיית ה CRITICAL SECTION ועומד בכל דרישות הפתרון



הפרוטוקול מבטיח קדימות של T2 על פני המתחרה בכניסה ל CRITICAL SECTION



שני תהליכים יכולים לשהות בו-זמנית ב CRITICAL SECTION



אף תשובה מבין המופיעות איננה נכונה

**משוב לבחירה בתשובה זו**

התשובה הנכונה:

הפרוטוקול מבטיח קדימות של T1 על פני המתחרה בכניסה ל CRITICAL SECTION

**שאלה 12**

תקין

10.00 נקודות מתוך 10.00

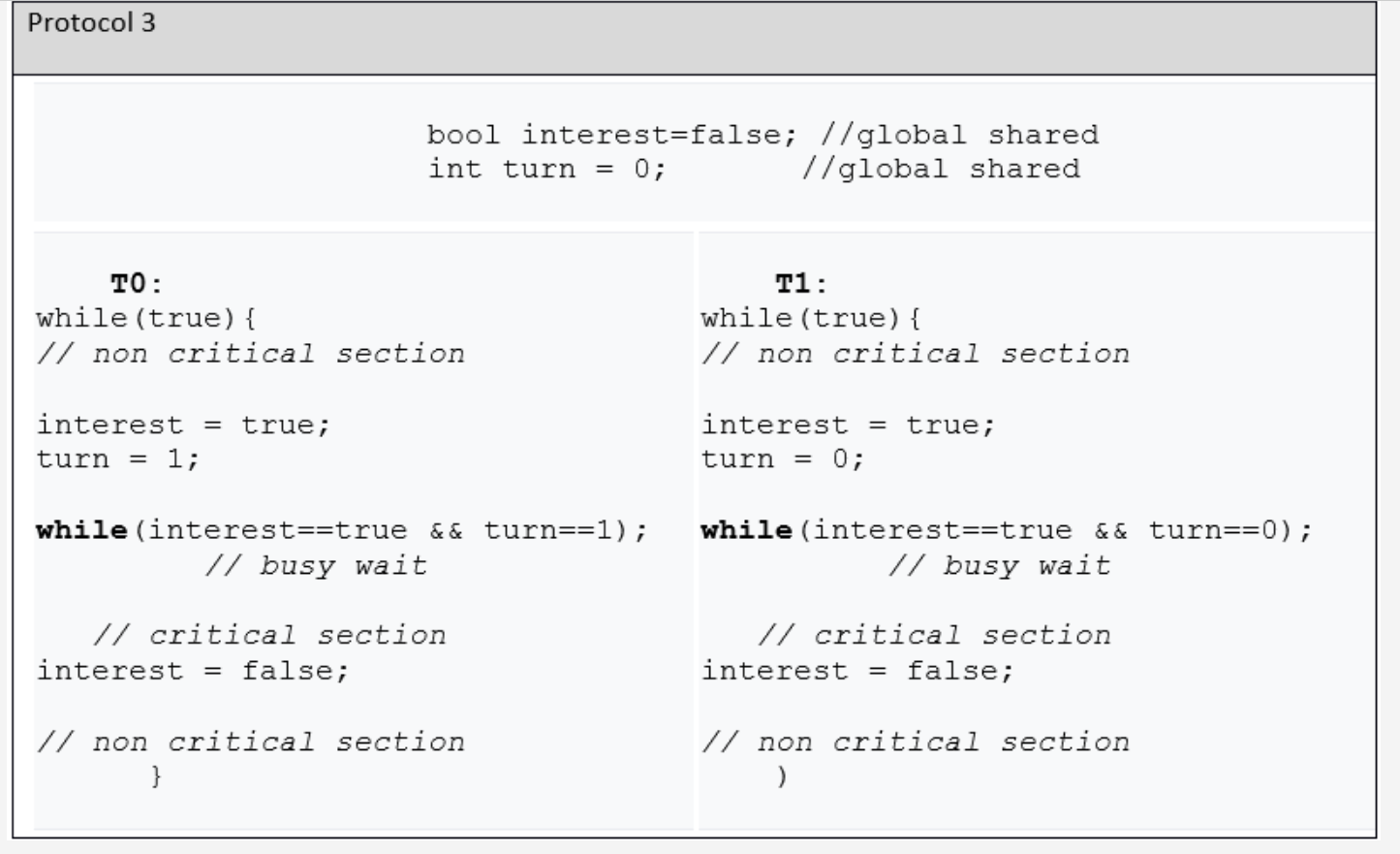
סימון שאלה

**תוכן השאלה**

**שאלה 8 (10%)**

נתונה התוכנית הבאה (pseudo code) שמיועדת ל  SYNCHRONIZATION בין **שני** THREADS T0 ו 1T שרצים במקביל. כל THREAD מבצע אותו פרוטוקול כניסה ויציאה ל  CRITICAL SECTION.

התוכנית משתמשת במשתנים גלובליים משותפים שנגישים ל 2 ה THREADS



בחרו את הטענה הנכונה:



הפרוטוקול פותר את בעיית ה CRITICAL SECTION



שני ה threads עלולים להיכנס למצב DEADLOCK



שני תהליכים יכולים לשהות בו-זמנית ב CRITICAL SECTION



הפרוטוקול אינו תקין ויכול לגרום לתקלה ברמת חומרה, בגלל שלשני ה THREADS יש אפשרות לגשת למשתנים משותפים בו-זמנית



הפרוטוקול לא מאפשר שיהיה בו-זמנית בקטע קריטי וגם מונע DEADLOCK, אבל הוא לא תקין כי THREAD מחוץ לקטע קריטי יכול למנוע מ THREAD אחר להיכנס אליו

**משוב לבחירה בתשובה זו**

התשובה הנכונה:

הפרוטוקול לא מאפשר שיהיה בו-זמנית בקטע קריטי וגם מונע DEADLOCK, אבל הוא לא תקין כי THREAD מחוץ לקטע קריטי יכול למנוע מ THREAD אחר להיכנס אליו

**שאלה 13**

הושלם

לא ניתן ציון

סימון שאלה

**תוכן השאלה**

**שאלה 8 - נימוק**

נמקי את בחירתך

נימוק עי פסילת השאר

1. רק פטרסון פותר את בעית הקטע הקריטי

2.הם לא יוכלו כי יש משתנה גלובאלי שתמיד מאפשר או לא או לו את התור

3. אין מצב כי תמיד התור שיך רק לאחד מהם

4.לא קשור בכלל אין בעיה ששני משתנים נגשים לאותו משתנה ברמת חומרה אולי ברמת תוכנה

**שאלה 14**

תקין

10.00 נקודות מתוך 10.00

סימון שאלה

**תוכן השאלה**

**שאלה 9 (10%)**

מערכת הקבצים של מערכת הפעלה מסוימת משתמשת בשיטת ה I-node עם פרמטרים הבאים:

* גודל הבלוק במערכת הקבצים הוא Kbytes 2
* כתובת הבלוק היא 4 בתים (bytes)
* 10 שדות של ה I-node יכולים להחזיק ישירות כתובת הבלוק בדיסק
* שדה נוסף אחד נועד להחזיק כתובת של ה single indirect block
* עוד שדה נוסף אחד נועד להחזיק כתובת של ה double indirect block
* ועוד שדה נוסף אחד נועד להחזיק כתובת של ה triple indirect block

חשבו מה הגודל המרבי של קובץ שניתן לאכסן ב 1514 בלוקים בסה"כ (כולל נתונים ומצביעים, אבל לא כולל את ה I-NODE  עצמו) .

הערה: לצורך החישוב  1 Mbyte = 1024 Kbyte , 1 Kbyte = 1024 byte



3020 Kbytes



3030 Kbytes



3008 Kbytes



3000 Kbytes



3028 Kbytes

**משוב לבחירה בתשובה זו**

התשובה הנכונה:

3020 Kbytes

**שאלה 15**

תקין

5.00 נקודות מתוך 5.00

סימון שאלה

**תוכן השאלה**

**שאלה 10 (5%)**

מה נכון לגבי HYPERVISOR מסוג 1 בתצורת bare metal?



כל משאבי החומרה hardware resources נשלטים **באופן ישיר** ע"י hypervisor



HYPERVISOR מסוג 1 בעצמו פועל ב USER MODE



רק הקצאות זיכרון ראשי עוברות דרך hypervisor, אבל פעולות קלט-פלט מבוצעות **ישירות** ע"י ה OS  האורחת **בלי שיתוף** של hypervisor



רק הקצאות זיכרון ראשי ופעולות קלט-פלט עוברות דרך hypervisor, אבל הקצאת זמן המעבד CPU מבוצעת **ישירות** ע"י ה OS האורחת **בלי שיתוף** של hypervisor



כאשר יישום המשתמש מבצע את system calls ל-OS שרצה על VMM , אותה ה OS **לא** **תנסה** לבצע את ה- system calls ב kernel mode

**משוב לבחירה בתשובה זו**

התשובה הנכונה:

כל משאבי החומרה hardware resources נשלטים **באופן ישיר** ע"י hypervisor

**שאלה 16**

תקין

5.00 נקודות מתוך 5.00

סימון שאלה

**תוכן השאלה**

**שאלה 11 (5%)**

מה נכון לגבי ניהול זיכרון ראשי (RAM) מדפדף (PAGING)?



זמני ביצוע גישות לכתובות וירטואליות VIRTUAL ADDRESSESS שונות תמיד אחידים



**בכל פניה** לזיכרון ראשי RAM יהיה צורך בתרגום כתובת וירטואלית VIRTUAL ADDRESS לכתובת פיזית PHYSICAL ADDRESS.



כתובת וירטואלית VIRTUAL ADDRESS תמיד באותה כתובת פיזית לכל אורך חיי התהליך



גודל הדף הווירטואלי תמיד צריך להיות שווה לגודל הדף הפיזי (מסגרת זיכרון פיזית), חוץ מהדף האחרון שיכול להיות יותר קטן



זמן טיפול בכל שגיאת דף בודדת PAGE FAULT תמיד אחיד

**משוב לבחירה בתשובה זו**

התשובה הנכונה:

**בכל פניה** לזיכרון ראשי RAM יהיה צורך בתרגום כתובת וירטואלית VIRTUAL ADDRESS לכתובת פיזית PHYSICAL ADDRESS.

**שאלה 17**

תקין

5.00 נקודות מתוך 5.00

סימון שאלה

**תוכן השאלה**

**שאלה 12 (5%)**

מה נכון לגבי מערכת הפעלה שמשתמשת ב COW – COPY ON WRITE כשיטת ניהול זיכרון ראשי RAM בזמן יצירת תהליך בן ע"י קריאת מערכת FORK לפני ביצוע קריאות מערכת נוספות?



בזמן יצירת תהליך הבן, כל מרחב הזיכרון של תהליך האב מועתק למרחב של הבן.



ברגע יצירת תהליך הבן, טבלת הדפים של תהליך האב מועתקת למרחב הזיכרון של הבן ולפני ביצוע פעלה כלשהי ע"י הבן או האב, שתי הטבלאות מצביעות על אותן כתובות פיזיות.



בזמן יצירת תהליך הבן שום דבר לא מועתק, שני התהליכים משתפים את אותה טבלת הדפים שנמצאת במרחב הזיכרון של האב כשלבן ניתנת גישה אליה. העתקת טבלת הדפים תתבצע בזמן PAGE FAULT אצל הבן.



חלק ממרחב הזיכרון של תהליך האב המתחיל מהנקודה שבה נוצר תהליך הבן מועתק למרחב הזיכרון של הבן.



איזור הקוד של תהליך האב יהיה מועתק למרחב הזיכרון של הבן, אבל איזור הנתונים לא מועתק ויישאר משותף לאב ולבן.

**משוב לבחירה בתשובה זו**

התשובה הנכונה:

ברגע יצירת תהליך הבן, טבלת הדפים של תהליך האב מועתקת למרחב הזיכרון של הבן ולפני ביצוע פעלה כלשהי ע"י הבן או האב, שתי הטבלאות מצביעות על אותן כתובות פיזיות.

**שאלה 18**

תקין

10.00 נקודות מתוך 10.00

סימון שאלה

**תוכן השאלה**

**שאלה 13 (10%)**

בהשוואה בין  pipeline ל boss-workers models , נתונים 8 threads שעובדים במקביליות אמיתית מלאה ו 11 הזמנות ייצור. את הביצוע של כל הזמנה ניתן לחלק לכמה שלבים שצריך.

ל boss-workers model לוקח 80ms לבצע הזמנה אחת, ל pipeline לוקח 10ms לכל שלב.

חשבו את הזמן הממוצע לביצוע הזמנה אחת עד לסיומה (כולל זמני המתנה) לכל מודל.



boss-workers  102  pipeline 130



boss-workers  80  pipeline  80



boss-workers  102  pipeline 180



boss-workers  160  pipeline 180



boss-workers  109  pipeline 130

**משוב לבחירה בתשובה זו**

התשובה הנכונה: boss-workers  109  pipeline 130

**שאלה 19**

שגוי

0.00 נקודות מתוך 5.00

סימון שאלה

**תוכן השאלה**

**שאלה 14 (5%)**

מהי הסיבה העיקרית לשימוש ב-­DMA (גישה ישירה לזיכרון) בביצוע פעולות קלט-פלט?



מאפשר להתקן, למשל דיסק, להשתמש ב RAM כמטמון.



הקטנת העומס על הזיכרון הראשי.



שיפור ביצועי המערכת ע"י הגדלת המקביליות



מאפשר לאותו התקן, למשל דיסק, לבצע כמה פעולות קלט-פלט במקביל.



למעבד CPU דרוש פחות זמן לביצוע **פקודה בודדת**.

**משוב לבחירה בתשובה זו**

התשובה הנכונה:

שיפור ביצועי המערכת ע"י הגדלת המקביליות

**שאלה 20**

תקין

5.00 נקודות מתוך 5.00

סימון שאלה

**תוכן השאלה**

**שאלה 15 (5%)**

על ידי מי ניתן  ליצור בפועל איזור של זיכרון  משותף SHARED MEMORY  בין שני תהליכים PROCESSES?



ע"י מערכת הפעלה בעקבות בקשת התהליך.



ע"י התהליך השני כשהוא ניגש לאזור השייך לתהליך הראשון.



תלוי במספר ליבות CORES שיש בתוך המעבד.



ע"י התהליך הראשון שניגש לאזור שעתיד להיות משותף.



תלוי בגודל האזור של זיכרון משותף

**משוב לבחירה בתשובה זו**

התשובה הנכונה:

ע"י מערכת הפעלה בעקבות בקשת התהליך.

**שאלה 21**

תקין

5.00 נקודות מתוך 5.00

סימון שאלה

**תוכן השאלה**

**שאלה 16 (5%)**

מה נכון בהשוואת 2 ארכיטקטורות של מחשוב הענן: Homogeneous  ו Heterogeneous ,עם 100 nodes שצריכים להריץ מספר משתנה של משימות**מאותו סוג**:



בעומס נמוך לא יהיה הבדל, אבל בעומס גבוה Heterogeneous תהיה יותר מהירה בגלל אפשרות של התאמה דינאמית במספר nodes המבצעים את העבודה



לא יהיה הבדל במהירות הביצוע בין 2 הארכיטקטורות



Heterogeneous תהיה יותר מהירה



בעומס נמוך לא יהיה הבדל, אבל בעומס גבוה Homogeneous תהיה יותר מהירה בגלל אפשרות של התאמה דינאמית במספר nodes המבצעים את העבודה



 Homogeneous תהיה יותר מהירה

**משוב לבחירה בתשובה זו**

התשובה הנכונה:

לא יהיה הבדל במהירות הביצוע בין 2 הארכיטקטורות