

בחינה בעקרונות מערכות הפעלה

קרא בעיון לפני שתתחיל בפתרון הבחינה!

- א. בבחינה זו 18 שאלות סגורות (מבחן אמריקאי).
עליך לבחור בכל פעם בתשובה יחידה מבין התשובות המוצעות ולהקיף בעיגול את אות התשובה שבחרת, על גבי השאלון עצמו בלבד ליד טקסט השאלה
- ב. הציון נקבע על פי מספר השגיאות. כל שגיאה מורידה 6 נקודות. כל שאלה ללא תשובה מורידה רק 4 נקודות. הציון ההתחלתי הוא 105. הציון המרבי שניתן לקבל הוא 100. הציון הנמוך ביותר שניתן לקבל הוא 0.
- ג. כל חומר עזר אסור לשימוש בזמן הבחינה פרט למחשבון כיס פשוט. הבחינה נמשכת שלוש שעות.
- ד. במקרה של ספק בהבנת ניסוח השאלה ניתן לכתוב על גבי שאלון הבחינה הערה המבהירה את דרך הבנתך של השאלה. אין לכתוב נימוקים לתשובה!
- בהצלחה!

שאלה 1.

פעולת TRAP במערכת הפעלה היא :

- (א) הפעולה הגורמת להגדלת ניצולת המעבד המרכזי.
- (ב) הפעולה הגורמת לנעילת סגמנטים (segments) בזיכרון הראשי.
- (ג) הפעולה הגורמת לנעילת מסגרות (frames) בזיכרון הראשי.
- (ד) אף תשובה קודמת אינה נכונה.

שאלה 2.

כידוע מעבד מרכזי (CPU) יכול להימצא באחד משני מצבים : מצב ראשוני (kernel mode) או מצב משתמש (user mode). קיום מצבים הללו נועד לאפשר מימוש של הגנה על פעילותה של מערכת ההפעלה. בחר/י מתוך הרשימה את מודל המבנה של מערכת ההפעלה (operating system architecture) בו (במודל שבחרת) אין שום אפשרות לממש הגנה כזו?

- (א) מודל מונוליטי (monolithic structure).
- (ב) מודל שרת-לקוח (client-server structure).
- (ג) מודל הקליפות (ring structure).
- (ד) מודל שרת-לקוח ומודל קליפות.
- (ה) אף תשובה קודמת איננה נכונה.

שאלה 3.

להלן טבלה עם זמני הגעת תהליכים והזמן הדרוש לעיבוד של כל אחד מהם. חשבו כמה תהליכים מסתיימים ביחידת זמן במוצע (throughput) כאשר אלגוריתם התזמון הוא SJF (shortest job first). מס' יחידות הזמן נמדד החל מתחילת עיבוד התהליך הראשון ועד לסיום התהליך האחרון.

| תהליך | זמן הגעה | זמן עיבוד נטו ב CPU |
|-------|----------|---------------------|
| A | 0 | 1 |
| B | 1 | 98 |
| C | 2 | 2 |
| D | 3 | 3 |

תזכורת: אלגוריתם SJF פועל כדלהלן – במידה ומגיעה עבודה (job) חדשה העבודה הנוכחית מושהית. והאלגוריתם בוחר מבין התהליכים הממתינים את התהליך הקרוב ביותר לסיומו.

- (א) 1/26.
- (ב) 1.
- (ג) 3.

שאלה 4.

לבעיית הקטע הקריטי עבור שני תהליכים P_i ו- P_j הוצע הפתרון הבא (כאשר המערכים $interested$ ו- $flag$ והמשתנה $turn$ משותפים לשני התהליכים) :

להלן הפרוטוקול עבור תהליך P_i .

```
...
interested[j] = FALSE;
flag[i] = TRUE;
while (interested[j] == TRUE)
{
    turn = j;
    interested[j] = FALSE;
}
turn = i;
while ( interested[j] == TRUE
        && turn == i
        && flag[j] == TRUE);
```

... /* Critical Section */

```
interested[j] = TRUE;
flag[i] = FALSE;
...
```

האם הפתרון פותר כיאות את בעיית הקטע הקריטי?

(א) כן.

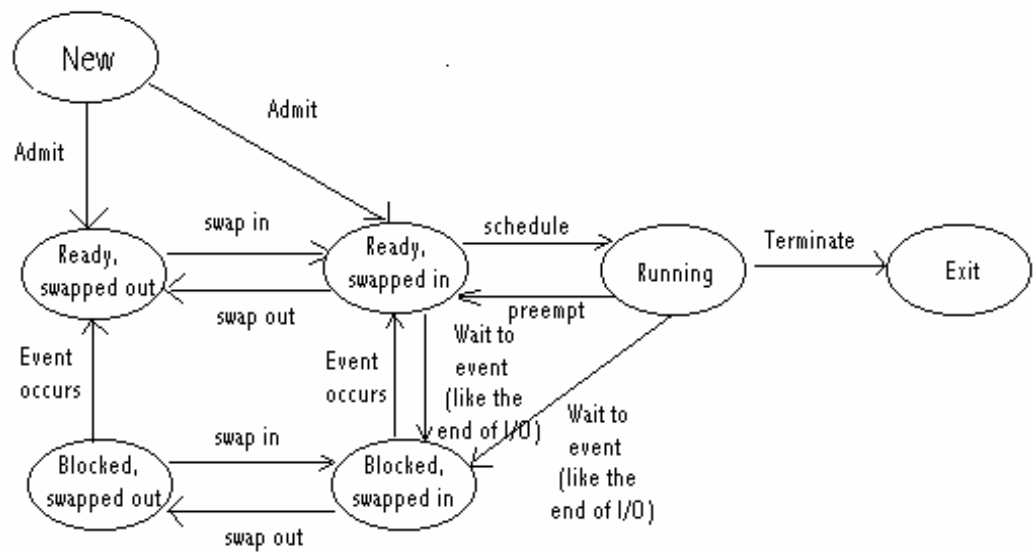
(ב) לא, כי שני תהליכים עלולים להימצא בו-זמנית בתוך קטע קריטי.

(ג) לא, כי תהליך מחוץ לקטע הקריטי עלול למנוע מתהליך אחר להיכנס לתוך קטע זה.

(ד) לא ניתן לקבוע לפי הנתון.

שאלה 5.

לגרף עם שלושת המצבים הבסיסיים של התהליך במערכת ההפעלה הוספנו עוד כמה מצבים:



עתה, קיים בגרף זה מעבר אחד מיותר. היכן נמצא מעבר זה?

(א) בין Running ל Ready, swapped in.

(ב) בין Ready, swapped in ל Blocked, swapped in.

(ג) בין New ל Ready, swapped out.

(ד) בין Running ו Exit.

שאלה 6.

מרחב הזיכרון הווירטואלי מכיל 8 דפים של 1024 מילים (words) לדף. הזיכרון הפיזי מכיל 32 מסגרות (frames). מהו מספר הסיביות (bits) המינימלי הדרוש לייצוג הכתובת הפיזית?

(א) 16.

(ב) 15.

(ג) 12.

(ד) 8.

שאלה 7.

במערכת המנהלת את הזיכרון ע"י דפדוף (paging), אבטחת המידע (כלומר מניעת מצב שבו תהליך אחד ניגש לזיכרון של תהליך אחר) נעשית כך:

(א) לכל תהליך יש טבלת דפים נפרדת.

(ב) מחלקים את מרחב הזיכרון המדומה לדפים כאשר לכל אחד מהדפים ישנה מחרוזת סיביות RWX (read, write, execute) אשר נועדה לקביעת רמת האבטחה של הדף.

(ג) ממקמים את הדפים במסגרות של הזיכרון הפיזי שמיועד ל buffer cache ומשתמשים במנגנון אבטחת המידע של buffer cache.

(ד) גורמים לפסיקה ומעבירים זאת לטיפולו של מתאם התקן (device driver) מתאים.

שאלה 8.

באלגוריתם MFU (most frequently used) להחלפת דפים בזיכרון מדופדף מחזיקים מונה של מספר פניות לכל דף. הדף המיועד להחלפה, בעת הצורך, יהיה הדף עם ערך המונה הגבוה ביותר. זאת איננה שיטה אופטימלית, אך ישנם נימוקים להפעלת השימוש בה. איזה טיעון, לדעתך, תומך בבחירת הדף על פי שיטה זו?

(א) MFU מהווה קירוב טוב לאלגוריתם האופטימלי.

(ב) האלגוריתם זול למימוש.

(ג) ייתכן כי הדף עם ערך מונה קטן הובא לזיכרון זה עתה וטרם זכה להתייחסות.

(ד) כל התשובות הקודמות נכונות.

שאלה 9.

מהו היתרון של שיטת דפדוף ללא טבלת הדפים (zero level paging):

- (א) פשטות מנגנון הדפדוף.
- (ב) מזעור מספר פסיקות הדף (page faults).
- (ג) הקטנת קבוצת העבודה (working set) של תהליך.
- (ד) כל התשובות הקודמות נכונות.

שאלה 10.

ברצונך ליצור קובץ במערכת UNIX כך שרק משה יכול לקרוא אותו, רק עליזה תוכל לכתוב אותו, ובנוסף שרק לך תהיה אפשרות גם לקרוא וגם לכתוב. הפתרון הוא:

- (א) הרשאת rw לך בתור בעל הקובץ, r לקבוצה (group) המכילה את משה, ו w לקבוצה המכילה את עליזה.
- (ב) הרשאת r לקבוצה המכילה את משה, והרשאת w לקבוצה המכילה אותך ואת עליזה.
- (ג) הרשאת rw לך בתור בעל הקובץ, w לקבוצה המכילה את עליזה, ו r לשאר המשתמשים.
- (ד) אין אפשרות ליצור הרשאות כאלה.

שאלה 11.

מהו הבדל בין ה RAM disk לבין ה buffer cache?

- א) למערכת ההפעלה אין שליטה על תוכן של ה RAM disk בזמן שיש לה שליטה על התוכן של ה buffer cache.
- ב) למערכת ההפעלה יש שליטה על תוכן של ה RAM disk בזמן שאין לה שליטה על התוכן של ה buffer cache.
- ג) ה buffer cache נמצא בזיכרון הראשי בזמן שה RAM disk נמצא בחוצץ הפנימי של דיסק.
- ד) ה buffer cache נמצא בחוצץ הפנימי של דיסק בזמן שה RAM disk נמצא בזיכרון הראשי.

שאלה 12.

מהו מס' פעולות קריאה/כתיבה של בלוק מהדיסק שצריך לבצע בהרצת התוכנית הבאה במערכת הפעלה UNIX כאשר:

- ניתן להניח כי הקובץ כבר קיים על הדיסק ומכיל מידע.
- גודלו של כל קובץ ספרייה היא בלוק אחד.
- הבלוקים (כולל ה super block) אינם נמצאים בזיכרון מראש.
- קריאות המערכת אינן נכשלות.
- מנגנון ה buffer cache מממש את מדיניות ה write through.
- ה i-nodes נמצאים בבלוקים שונים.

```
fd = open("/a/b/c");  
write(fd, &buf, 1);
```

רמז: כיצד ניתן לדעת מהו מספר ה i-node של קובץ הספרייה "/".

א) 5.

ב) 8.

ג) 9.

ד) 10.

שאלה 13.

תכנית מבקשת ליצור תהליך חדש נוסף (למשל ע"י קריאת המערכת fork) אבל טבלת התהליכים מלאה. במקרה כזה:

- (א) מערכת ההפעלה חייבת להרוג את התהליך המבקש מייד.
- (ב) מערכת ההפעלה צריכה להרוג תהליך כלשהו עם עדיפות נמוכה ולרשום את התהליך החדש במקום שהתפנה בטבלת התהליכים.
- (ג) מערכת ההפעלה צריכה להכשיל את הפעולה כי אחרת טבלת התהליכים תישאר במצב לא קונסיסטנטי.
- (ד) מערכת ההפעלה צריכה להכשיל את הפעולה כי אם התהליך ימתין בתקווה שתתפנה כניסה כלשהי בטבלת הדפים, עלול להיווצר deadlock.

שאלה 14.

אלגוריתם הבנקאי להתחמקות מהקיפאון משתמש במבני נתונים הבאים :

E הוא מערך חד-ממדי בגודל m . בכל תא $E(j)$ מאוחסן מספר המשאבים הכללי במערכת השייכים למחלקה j .

A הוא מערך חד-ממדי בגודל m . בכל תא $A(j)$ מאוחסן מספר המשאבים החופשיים (שלא הוקצו עדיין) מהמחלקה j .

P הוא מערך חד-ממדי בגודל m . בכל תא $P(j)$ מאוחסן מספר המשאבים התפוסים (אשר הוקצו) מהמחלקה j .

C הוא מערך דו-ממדי בגודל $n*m$. בתא $C(i)(j)$ מאוחסן מספר המשאבים מהמחלקה j שכבר הוקצו לתהליך i .

R הוא מערך דו-ממדי בגודל $n*m$. בתא $R(i)(j)$ מאוחסן מספר המשאבים מהמחלקה j הנדרשים ע"י התהליך i במהלך כל חייו.

האם ניתן באמצעות האלגוריתם לזהות מהי קבוצת התהליכים הכלואים בקיפאון?

(א) כן.

(ב) לא ניתן, מפני שהאלגוריתם מזהה את קיום מצב הקיפאון במערכת כולה ולא לגבי קבוצת תהליכים מסוימת.

(ג) לא ניתן, מפני ששיטת השימוש באלגוריתם לא מאפשרת כניסה למצב הקיפאון.

(ד) לא ניתן, מכיון שלא קיימת המתנה מעגלית.

שאלה 15.

מה יכולה להיות הסיבה לכך שתהליך בן במערכת UNIX לא "יורש" מתהליך אב את מצב פסיקות התוכנה (signal disposition) של האב בעקבות הקריאה לקריאת המערכת `fork`?

(א) למשל, כאשר הדבר נעשה כדי לאפשר לתהליך הבן "להתעלם" מקבלת הסיגנל `SIGKILL`.

(ב) למשל, כאשר הדבר נעשה כדי לגרום לקביעת מצב פסיקות התוכנה לטיפול בררת המחדל.

(ג) טענת השאלה אינה נכונה היות ותהליך הבן אכן "יורש" את מצב פסיקות התוכנה של האב.

(ד) 2 תשובות מבין התשובות הקודמות הן נכונות.

תזכורת: קביעת המצב של פסיקות התוכנה נעשה בעזרת קריאת המערכת `signal`.

שאלה 16.

מהו `major device number` שמכיל ה `i-node` של קובץ מיחד (special file) במערכת UNIX?

- (א) מספר לזיהוי מתאם התקן (device driver).
- (ב) מספר לזיהוי התקן , כגון דיסק מסוים מבין דיסקים רבים.
- (ג) מזהה מערכת קבצים אשר ה i-node שייך לה.
- (ד) אף תשובה קודמת אינה נכונה.

שאלה 17.

כמה תהליכים ייווצרו בעקבות ההרצה של התוכנית הבאה כאשר ניתן להניח הצלחת כל קריאות המערכת.

```
pid_1 = fork();
pid_2 = fork();
pid_3 = fork();
```

- (א) 6.
- (ב) 7.
- (ג) 8.
- (ד) 9.
- (ה) 10.

הערה : כולל התהליך שמבצע את הקוד.

שאלה 18.

איך ממומש מנגנון ההגנה של מערכת הקבצים ב NT?

- (א) באמצעות רשימת היכולות (C –lists, capability lists).
- (ב) באמצעות רשימת גישה (ACL-Access control list).
- (ג) באמצעות מטריצת ההגנה (protection matrix).
- (ד) אף תשובה קודמת אינה נכונה.

בהצלחה