

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

המושג standby page במנגנון ניהול הזיכרון באמצעות הדפדוף של מערכת ההפעלה NT מתייחס אל:

- הדף אשר נמצא בקבוצת העבודה (working set) של התהליך
- הדף אשר הוצאה מקבוצת העבודה של התהליך אך לא פונה ממסגרתו בזיכרון הפיזי
- הדף אשר כל גישה אליו תגרום לפסיקת דף (page fault)
- הדף אשר אינו נמצא בזיכרון הפיסי

שאלה 2

המושג MFT (master file table) מתייחס למערכת קבצים מסוג NTFS של מערכת ההפעלה NT. מהי MFT?

- טבלה המכילה מידע לגבי הבלוקים הנמצאים ב-buffer cache. (כגון כתובת הבלוק בדיסק, מידע על אלגוריתם ניהול ה-cache וכו')
- קובץ המורכב מרשומות (records) בגודל קבוע אשר מכילות מס' מאפיינים (attributes) של כל אחד מהקבצים ב-NTFS
- קובץ המכיל קוד לטעינת מערכת ההפעלה (startup code)
- אף תשובה אינה נכונה

שאלה 3

אחד ממדדי אלגוריתם התזמון הוא זמן שהיה (turnaround time) שנסמן ב T_q . אך מדד זה אינו משקף את זמן ההמתנה של התהליך ביחס לתהליכים אחרים. לכן הוצע מדד חדש: זמן שהיה מנורמל (normalized turnaround time) שנסמן ב T_n . מחשבים את T_n באופן הבא: $T_n = T_q / T_s$ כאשר T_s הנו זמן CPU. חשבו בקירוב את זמן שהיה המנורמל הממוצע עבור התהליכים הבאים במידה ואלגוריתם התזמון הוא FIFO.

התהליך	זמן ההגעה	זמן CPU
A	0	1
B	1	100
C	2	1
D	3	100

- 100
- 26
- 1.99
- 38

שאלה 4

מערכת ההפעלה NT מעלה את עדיפותו של thread ברגע שהוא חוזר מהמצב wait. עבור איזה מה threads הבאים ערך העלאת העדיפות יהיה גבוה יותר?

- א. עבור idle thread
- ב. עבור ה thread שהמתין לסיום פעולת קלט/פלט עם הדיסק
- ג. עבור ה thread שהמתין לקלט מהמקלדת
- ד. בשני המקרים (א) ו-(ב) הערך יהיה זהה

שאלה 5

מה **לא** יכול להתרחש בעקבות קריאה לקריאת המערכת wait במערכת ההפעלה UNIX?

- א. התהליך הקורא ל wait מושהה באופן מיידי
- ב. הקריאה חוזרת מייד עם הסטטוס של אחד מתהליכי הבנים אשר הסתיימו
- ג. במידה ואחד מבניו של התהליך אשר קרא לwait הסתיים, מתפנית כניסה (entry) בטבלת התהליכים של המערכת (process table)
- ד. אף תשובה אינה נכונה

שאלה 6

להלן רשימת פעולות I/O. מה מבין הרשימה **לא בסמכותה** של שכבת device independent I/O של תוכנת קלט/פלט?

- א. ניהול הקצאת הבלוקים בדיסק
- ב. buffering (ניהול של ה buffer cache)
- ג. הגנה על התקנים (device protection)
- ד. DMA (direct memory access)

שאלה 7

אחד מיתרונותיו של האלגוריתם SSF (shortest seek first) לתזמון זרוע הדיסק הוא?

- א. הגינות (fairness)
- ב. העדפת מסלולים פנימיים יותר
- ג. מסי מסלולים מרבי שמטופל ביחידת זמן
- ד. אף תשובה אינה נכונה

שאלה 8

מערכת UNIX מספקת דרכים להגנה על משאביה ועל התקני החומרה. רק משתמשים מורשים או תהליכים המורצים ע"י המשתמשים המורשים יכולים לגשת אליהם. אך לעתים נוצרים מצבים בהם משתמשים רגילים צריכים גישה לביצוע פעולות מסוימות. לכן UNIX מספקת מנגנונים המאפשרים למשתמשים רגילים לבצע פעולות הללו. מה מספקת UNIX בשכבה ?device independent I/O level

- א. setuid bit במחרוזת הרשאות הקבצים
- ב. שדים (daemons) שרצים ברקע ומקבלים בקשות קלט/פלט
- ג. wrx bits במחרוזת הרשאות הקבצים
- ד. symbolic links

שאלה 9

באיזו מארבעת הרמות של תוכנת קלט/פלט מתבצעת ההגנה על התקני חומרה במערכת UNIX?

- א. במערכת הטיפול בפסיקות שיוצרים ההתקנים (interrupt handling mechanism)
- ב. בתוך תוכנת התיאום בין ההתקן לבין מערכת ההפעלה (device drivers)
- ג. בתוך תוכנת קלט/פלט הבלתי תלויה בהתקן (device independent software)
- ד. בשדים (daemons) לארגון הפלט ופונקציות ספריה שאינן תלויות חומרה

שאלה 10

מהו interrupt vector?

- א. טבלה המכילה כתובות של הפונקציות האמורות להתבצע בעקבות פסיקות תוכנה (signals) במערכת UNIX
- ב. טבלה המכילה כתובות של device drivers לכל סוג של פסיקת חומרה (interrupt)
- ג. טבלה המכילה כתובות של interrupt handler routine לכל סוג של פסיקת חומרה (interrupt)
- ד. שתיים מהתשובות נכונות

שאלה 11

תפקידו של Paging Daemon הוא :

- א. למפות כתובת וירטואלית לכתובת פיזית
- ב. להעלות תהליך או חלק ממנו מהדיסק לזיכרון
- ג. לנהל את סדר הגעת התהליכים לזיכרון
- ד. לדאוג למאגר דפים פנויים בזיכרון המדופדף

שאלה 12

מה המידע המצוי בספרייה (Directory) של מערכת הקבצים במערכת UNIX ?

- א. שמות הקבצים ולכל קובץ מספר ה-i-node שלו
- ב. שמות הקבצים ולכל קובץ מספר הבלוק הראשון שלו
- ג. שמות הקבצים ולכל קובץ התכונות (Attributes) שלו
- ד. שמות בקבצים ולכל קובץ מספר ה-i-node והתכונות (Attributes) שלו
- ה. שמות הקבצים ולכל קובץ מספר הבלוק הראשון והתכונות (Attributes) שלו

שאלה 13

להלן קטע קוד של התהליך המתבצע בסביבת מערכת הפעלה UNIX :

```
pid = fork();
if (pid == 0)
    printf("in child\n");
else
    printf("in parent\n");
```

בהנחה שקריאת המערכת fork הצליחה, הייתכן שהתהליכים (המקורי ותהליך הבן הנוצר) יהיו פרק זמן מסוים במצב zombie?

- א. לא ייתכן
- ב. ייתכן ותהליך הבן בלבד
- ג. ייתכן ותהליך האב בלבד
- ד. ייתכן ושניהם

שאלה 14

מהו הפלט האפשרי אשר יודפס על המסך ע"י התוכנית הבאה :

```
main() {
    while (fork() != 0)
        printf("1");
}
```

- א. 1
- ב. סדרה ארוכה אך באורך סופי של 1-ים
- ג. סדרה של N-1 ימים כאשר N הוא מספר התהליכים המקסימלי המותר למפעיל התכנית
- ד. התכנית לא תדפיס דבר
- ה. סדרה אין סופית של 1-ים
- ו. אף תשובה קודמת איננה נכונה

שאלה 15

האם ניתן לאפשר, ששני תהליכים שונים, ישתמשו יחד בו-זמנית באותה קבוצה משותפת של דפים בזיכרון?

- א. כן. רק בתנאי שהתהליכים מריצים את אותה תוכנית בדיוק
- ב. כן. רק בתנאי שכל הדפים הם לקריאה בלבד
- ג. כן. אבל במצב כזה המערכת חייבת ליצור שני העתקים נפרדים של כל הדפים המשותפים
- ד. כל התשובות נכונות

שאלה 16

במערכת קבצים נתונה משתמשים בשיטת i-nodes. נתון: גודל הבלוק הוא 0.5K, גודל המצביע לבלוק הוא 8 בתים. כל i-node מכיל 10 מצביעים ישירים, ושלושה מצביעים לא ישירים: single-indirect, double-indirect ו-triple-indirect (מצביע אחד מכל סוג). ידוע כי קובץ מסוים משתמש ב-144 בלוקים (כולל את הבלוקים של הנתונים ואת הבלוקים של המצביעים פרט ל-i-node עצמו).

מהו גודלו של הקובץ הזה ?

- א. 70K
- ב. 70.5K
- ג. 72K
- ד. 140K
- ה. 142K
- ו. 144K

שאלה 17

בחרו שורה מן הטבלה המכילה טענה נכונה הן לגבי פתרון פטרסון (Peterson) והן לגבי פתרון ע"י הוראת TSL (test and set lock) לבעיית הקטע הקריטי:

פתרון פטרסון		TSL
א.	פותר כיאות את בעיית הקטע הקריטי	דורש ידע על מהירות ביצוע התהליכים
ב.	גורם להמתנה פעילה	דורש תמיכה מיוחדת בחומרה
ג.	דורש ידע על מהירות ריצתם של התהליכים	דורש חסימת פסיקות (interrupts)
ד.	מפר את התנאי של מניעה הדדית (mutual exclusion)	פותר כיאות את בעיית הקטע הקריטי

שאלה 18

בבעיית למפורט (Lamport's bakery problem) מדובר במאפיה ובה N מוכרים. כל קונה הנכנס למאפיה לוקח מספר ומחכה עד שמגיע תורו ואז ניגש לאחד מהמוכרים הפנויים. ברגע שמוכר מתפנה הוא קורה ללקוח הבא התור. להלן הצעת פתרון לבעיית למפורט בהנחה שכל לקוח שלקח מספר מחכה עד אשר מגיע תורו ולא עוזב את המאפיה.

```

int curr_customer = 1;          /* customer about be called by some salesman */
int number = 1;                 /* numbers picked by customers */
semaphore number_mutex = 1;     /* controls access to the number_mutex */
semaphore curr_customer_mutex = 1; /* controls access to the curr_customer_mutex */

void salesman (void){
    while(1) {
        down(number_mutex);      /* get exclusive access to number */
        down(curr_customer_mutex); /* get exclusive access to curr_customer */
    }
}

```

```

    if (number >= curr_custmer)           /* if there are waiting customers */
        curr_customer = curr_customer + 1; /* increase to indicate availability */
    up(number_mutex);                      /* release number */
    up(curr-customer_mutex);               /* release customer */
    serve_the_customer();                  /* non-critical section */
}
}

void customer(void){
    int my_turn;                          /* indicates customer's place in the line */
    down(number_mutex);                   /* get exclusive access to the number */
    my_turn = number;
    number = number + 1;
    up(number_mutex);                     /* release access to number*/
    get_the_salesman_when(my_turn); /*non-critical section*/
}

```

האם פתרון זה מונע את היווצרותו של מצב הקיפאון (deadlock)?

- א. הפתרון מונע את היווצרות מצב הקיפאון
- ב. הפתרון היה מונע את היווצרותו של מצב הקיפאון במידה והיו מאתחלים את ערכם של שני הסמפורים ב-0
- ג. הפתרון היה מונע את היווצרותו של מצב הקיפאון במידה והיו מחליפים סדר בין down(number_mutex) לבין down(curr_customer_mutex)
- ד. הפתרון אינו מונע את היווצרות מצב הקיפאון

סוף