

## בחינה בעקרונות מערכות הפעלה

קרא בעיון לפני שתתחיל בפתרון הבחינה!

- א. בבחינה זו 20 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 התשובות הנכונות. כל תשובה נכונה מזכה ב 5 נקודות.
- ג. כל חומר עזר אסור לשימוש בזמן הבחינה פרט למחשבון 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!

## חינה 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 5 נקודות. כל שאלה ללא תשובה מורידה רק 3 נקודות. הציון ההתחלתי הוא 100. הציון המרבי שניתן לקבל הוא 100. הציון הנמוך ביותר שניתן לקבל הוא 0.
- ג. כל חומר עזר אסור לשימוש בזמן הבחינה פרט למחשבון 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

אילו מהטבלאות הבאות מוחזקות עבור כל תהליך בנפרד?

(א) טבלת דפים רגילה (page table)

(ב) טבלת דפים מהופכת (inverted page table)

(ג) טבלת תהליכים (process table)

(ד) כל התשובות הקודמות הן נכונות

## שאלה 2

לפניכם מספר טענות הקשורות למצבי קיפאון (deadlock). בחרו בתשובה נכונה:

(א) בקיפאון מעורבים לפחות שני תהליכים/תהליכונים (threads/processes)

(ב) אלגוריתם הבנקאי (banker's algorithm) מונע היווצרות מצבי קיפאון

(ג) לולאה אינסופית (endless loop) הנה סוג של מצב קיפאון

(ד) הטענות א ו ב הן הנכונות

(ה) הטענות ב ו ג הן הנכונות

## שאלה 3

האם קטע קוד הבא מהווה קטע קריטי במידה ושנים או יותר משני תהליכים מבצעים insert אל אותו התור? אפשר להניח כי כל שורה מתבצעת באופן אטומי (כלומר התהליך אינו מתחלף).

```
struct node{
    data;
    node_ptr next;
}
```

```
struct queue{
    node_ptr first;
}
```

```
insert(node_ptr, queue){ /* node_ptr is a pointer to the allocated and
                           initialized node is about inserted into the queue */
    tmp_ptr = queue->first;
    queue->first = node_ptr;
    node_ptr->next= tmp_ptr;
}
```

(א) כן. מכיוון שהתהליכים המבצעים את קטע הקוד הזה עלולים להיקלע למצב הקיפאון (deadlock)

(ב) כן. מכיוון שהתור עלול להשתבש כתוצאה מהרצת קטע קוד זה ע"י שניים או יותר תהליכים

(ג) לא. מכיוון וכל שורה מתבצעת כפעולה אטומית

(ד) לא ניתן לענות על השאלה על סמך הנתונים הנ"ל

## שאלה 4

מהו החסרון העיקרי של האלגוריתם להחלפת דפים (Not Frequently Used) NFU?

- (א) האלגוריתם לא ניתן למימוש בחומרה
- (ב) האלגוריתם מקפח תהליכים אינטראקטיביים
- (ג) האלגוריתם לוקח בחשבון את המספר האבסולוטי של פניות לדפים מאז הרצת התהליך
- (ד) האלגוריתם לוקח בחשבון רק את הפניות אשר בוצעו במשך פרק זמן מסוים

## שאלה 5

לפניכם רשימת זוגות. בין אלו מזוגות המדדים קיימת תלות כך ששיפור באחד מהם גורם לנסיגה בשני (trade-off). בחרו בטענה הנכונה ביותר:

- (א) הגינות (fairness) ואיזון העומסים במערכת (balance)
- (ב) זמן תגובה (response time) וניצולת CPU (CPU utilization)
- (ג) זמן שהיה (turnaround time) וקצב סיום עבודות (throughput)
- (ד) התשובות א' ו ב' הן נכונות
- (ה) התשובות א', ב' וג' הן נכונות

## שאלה 6

באיזו שכבה של תוכנת קלט/פלט מתבצע חישוב המסלול (track), הסקטור, והראש הקורא במקרה של קריאה מדיסק?

- (א) במערכת הטיפול בפסיקות הנוצרות ע"י 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 7

מה הן התכונות של האלגוריתם lottery scheduling?

- (א) האלגוריתם מבטיח שאם תהליך מגיע ראשון הוא גם ירוץ ראשון (first come first served)
- (ב) האלגוריתם מבטיח זמן שהיה (turnaround time) מזערי במערכת
- (ג) האלגוריתם שואף לחלק את זמן החישוב (CPU) בין משתמשי מערכת (או תהליכי מערכת) במידה שווה
- (ד) אף תשובה קודמת איננה נכונה

## שאלה 8

נתונה מערכת אשר מנהלת זיכרון באמצעות הדפדוף (paging). להלן תוכן טבלת TLB (Translation Lookaside Buffer) במהלך ריצתו של תהליך מסוים.

מס' דף	valid bit	modified bit	מס' מסגרת
0	1	0	4
1	1	1	7
2	1	1	3
3	0	1	-

לאיזו כתובת פיזית מתייחסת כתובת המדומה 1052 אם ידוע כי כל המספרים מוצגים בשיטה העשרונית, גודל הדף הנו 1024 בתים (bytes), מילת הזיכרון הנה בית אחד.

(א) 7196

(ב) 5497

(ג) 6072

(ד) 4291

## שאלה 9

פונקציה exec במערכת UNIX משמשת ל:

(א) טעינת תהליך חדש

(ב) שינוי גודל סגמנט ה data

(ג) יצירת קשורים סימבוליים (symbolic links)

(ד) לקריאה/כתיבה מתוך/אל קבצים

## שאלה 10

איזה מבין האלגוריתמים הבאים להחלפת דפים אינו אלגוריתם מחסנית (stack algorithm) ?

(א) NRU (Not Recently Used)

(ב) FIFO (First In First Out)

(ג) second chance

(ד) כל האלגוריתמים ברשימה אינם מסוג אלגוריתם מחסנית

## שאלה 11

צעד אחד של אלגוריתם לעיבוד תמונה מחלק את התמונה לחלקים ומבצע עיבוד על חלקיה באמצעות מספר תהליכים מקביליים. על האלגוריתם לבצע  $N$  צעדים כאשר נתון כי לא ניתן להמשיך לצעד הבא עד שכל התהליכים סיימו את העיבוד בצעד הקודם. מהו מנגנון הסנכרון שתציעו לסנכרון התהליכים של האלגוריתם המקבילי?

- (א) סמפורים (semaphores)
- (ב) מוניטורים (monitors)
- (ג) העברת הודעות (message passing)
- (ד) מחסומים (barriers)

## שאלה 12

מערכת הקבצים (file system) של UNIX נמצאת במצב לא עקבי (inconsistent state) והגורם לכך הוא בלוק חסר (missing block). כיצד ניתן להחזיר את מערכת הקבצים למצב עקבי?

- (א) להוסיף את הבלוק החסר לרשימת הבלוקים הפנויים
- (ב) להסיר את הבלוק החסר מרשימת הבלוקים הפנויים
- (ג) לא ניתן להחזיר את מערכת הקבצים למצב עקבי
- (ד) לשכפל את הבלוק החסר

## שאלה 13

לפניכם מספר טענות בנוגע לשכפול של file descriptor במערכת הפעלה UNIX. בחרו בטענה הנכונה ביותר:

- (א) במערכת הפעלה UNIX הקריאה ל fork יכולה לשכפל את ה file descriptor של קובץ פתוח
- (ב) במערכת הפעלה UNIX הקריאה ל dup יכולה לשכפל את ה file descriptor של קובץ פתוח
- (ג) שתי התשובות הקודמות הן נכונות
- (ד) אף תשובה קודמת איננה נכונה

## שאלה 14

בחרו שורה מן הטבלה המכילה טענה נכונה גם לגבי סיגנלים (signals) וגם לגבי פסיקות חומרה (interrupts) במערכת ההפעלה UNIX:

פסיקות	סיגנלים	
נוצרות ע"י התקני חומרה בלבד	נשלחים ע"י תהליכים לתהליכים אחרים	א.
נוצרות כתוצאה מקריאה ל system calls בלבד	נשלחים לתהליכים ע"י CPU ישירות	ב.
נוצרות ע"י interrupt handler routine	נשלחים כתוצאה מקריאות מערכת ( system calls ) בלבד	ג.
נוצרות ע"י שעון	נשלחים ע"י מתאמי התקנים (device drivers) בלבד	ד.

## שאלה 15

אחד החסרונות המובהקים של שיטת ה I-nodes היא :

- (א) ריסוק פנימי (internal fragmentation) משמעותי של שטח הדיסק
- (ב) השיטה אינה מאפשרת גישה אקראית (random access) לקובץ
- (ג) מספר גבוה של גישות לדיסק בזמן עבודה עם קבצים גדולים
- (ד) אף תשובה קודמת איננה נכונה

## שאלה 16

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

- (א) הגינות (fairness)
- (ב) העדפת מסלולים פנימיים יותר
- (ג) מספר מסלולים מירבי שמטופל בכל יחידת זמן
- (ד) אף תשובה קודמת אינה נכונה

## שאלה 17

לפניכם מספר טענות הקשורות למערכת קלט/פלט. בחרו בטענה הנכונה ביותר :

- (א) מנגנון DMA (Direct Memory Access) מקטין את מספר פסיקות הקלט/פלט (I/O interrupts) בהשוואה לשיטת interrupt-driven I/O
- (ב) אלגוריתם SSF (Shortest Seek First) לתזמון זרוע דיסק מונע הרעבה (starvation)
- (ג) מתאם ההתקן (device driver) של דיסק מורשה להשתמש בקריאת מערכת write של מערכת הפעלה על מנת לייעל את ביצועיו
- (ד) כל התשובות הקודמות הן נכונות

## שאלה 18

כאשר מדובר על מבנה מערכת הפעלה לפי מודל שרת-לקוח (client-server model) מהי התכונה אשר מהווה חסרון מובהק של המודל?

- (א) העדר מבנה כלשהו. המערכת היא אוסף שגרות אשר כל אחת מהן יכולה לקרוא לשגרה אחרת מן האוסף
- (ב) חוסר אפשרות התאמה למערכות מבוזרות (distributed systems)
- (ג) תקורת (overhead) התקשורת בין רכיבי המערכת
- (ד) כל התשובות הקודמות הן נכונות

**בהצלחה!**

## חינה 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 5 נקודות. כל שאלה ללא תשובה מורידה רק 3 נקודות. הציון ההתחלתי הוא 100. הציון המרבי שניתן לקבל הוא 100. הציון הנמוך ביותר שניתן לקבל הוא 0.
- ג. כל חומר עזר אסור לשימוש בזמן הבחינה פרט למחשבון 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

להלן סדרת פעולות הננקטות לאחר פסיקת דף (page fault):

- (1) ביצוע של פעולת ה TRAP למערכת הפעלה
- (2) שמירת האוגרים (registers) ושמירת מצב התהליך (process state)
- (3) אימות הרשאות גישה לדף
- (4) הפעלת אלגוריתם לחילוף הדפים (page replacement algorithm) במידת הצורך
- (5) הקצאת CPU לטובת תהליך אחר, אשר מוכן לריצה
- (6) תחילת העברת הדף מהדיסק למסגרת הפנויה
- (7) עדכון טבלת הדפים ומבני נתונים נוספים להימצאות הדף בזיכרון הראשי
- (8) עם הקצאת ה CPU לטובת התהליך שגרם לפסיקת דף, טעינת האוגרים השמורים ושחזור מצב התהליך
- (9) המשך ביצוע מהפקודה אשר גרמה לפסיקת דף

האם סביר לנקוט באחד הצעדים הבאים על מנת לשפר, באופן כללי, את ביצועי המערכת. בחרו בתשובה נכונה:

- (א) יש צורך לשנות את הסדר ולבצע צעד 3 לאחר הצעד 6
- (ב) יש צורך לשנות את הסדר ולבצע צעד 1 לאחר הצעד 3
- (ג) יש צורך לשנות את הסדר ולבצע צעד 7 לאחר הצעד 8
- (ד) סדרת הפעולות הננקטת לאחר פסיקת דף הנה סדרה סבירה באופן כללי

## שאלה 2

לפניכם מספר טענות הקשורות למצבי קיפאון (deadlock). בחרו בתשובה נכונה:

- (א) בקיפאון מעורבים לפחות שני תהליכים/תהליכונים (threads/processes)
- (ב) אלגוריתם הבנקאי (banker's algorithm) מונע היווצרות מצבי קיפאון
- (ג) לולאה אינסופית (endless loop) הנה סוג של מצב קיפאון
- (ד) הטענות א ו ב הן הנכונות
- (ה) הטענות ב ו ג הן הנכונות

### שאלה 3

לפניכם פסאודו-קוד של הוראת מחשב הכתובה בסגנון של שפת C:

```
int add(int& common, int value){  
    int tmp = common;  
    common += value;  
    return (tmp);  
}
```

ההוראה מתבצעת באופן **אטומי** כאשר הפרמטר הראשון של ההוראה מועבר אליה by reference והפרמטר השני מועבר by value.

עיינו בפסאודו-קוד של פרוטוקול להגנה על קטע קריטי שמשותף לשני תהליכים:

Process 0	Process 1
<pre>while (1){     while (add(common_var, 1) != 0)         add(common_var, -1);     &lt;Critical Section&gt;     add(common_var, -1); }</pre>	<pre>while (1){     while (add(common_var, 1) != 0)         add(common_var, -1);     &lt;Critical Section&gt;     add(common_var, -1); }</pre>

המשתנה common\_var הוא משותף לשני תהליכים ומאותחל ל 0. בחרו בתשובה נכונה לגבי הפרוטוקול הנ"ל:

- (א) שני התהליכים יכולים להימצא בו זמנית בתוך הקטע הקריטי
- (ב) אחד מהתהליכים יכול להישאר מחוץ לקטע קריטי ללא יכולת להיכנס
- (ג) הפרוטוקול הנ"ל מהווה פתרון סביר לבעיות המרוץ (race condition problems) למרות שהוא משתמש בהמתנה פעילה (busy waiting)
- (ד) כל תשובות הקודמות הן נכונות

### שאלה 4

בחרו בטענה הנכונה לגבי השימוש ב mutex:

- (א) השימוש ב mutex גורם להפרת התנאי של מניעה הדדית (mutual exclusion)
- (ב) השימוש ב mutex מאפשר ספירה (counting)
- (ג) mutex יעיל לשימוש ברמת המשתמש
- (ד) אף תשובה קודמת איננה נכונה

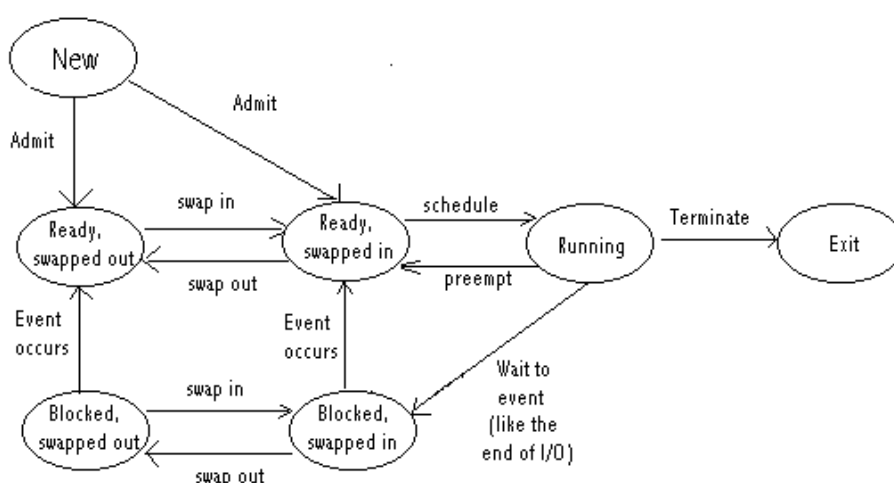
## שאלה 5

לפניכם רשימת זוגות. בין אלו מזוגות המדדים קיימת תלות כך ששיפור באחד מהם גורם לנסיגה בשני (trade-off). בחרו בטענה הנכונה ביותר :

- (א) הגינות (fairness) ואיזון העומסים במערכת (balance)
- (ב) זמן תגובה (response time) וניצולת CPU (CPU utilization)
- (ג) זמן שהיה (turnaround time) וקצב סיום עבודות (throughput)
- (ד) התשובות א' ו ב' הן נכונות
- (ה) התשובות א', ב' וג' הן נכונות

## שאלה 6

לפניכם מכונת המצבים של מתזמן במערכת הפעלה UNIX :



מדוע, לדעתך, אין מעבר בין המצב *blocked, swapped in* למצב *running*?

- (א) הוספת מעבר כזה איננה חוקית מכיון שעל מנת להיות *running* התהליך חייב לעבור דרך אחד ממצבי *ready*
- (ב) הוספת מעבר כזה מיותרת מכיון שהתהליכים במצב *blocked, swapped in* בלאו הכי לא יגיעו למצב *running*
- (ג) הוספת מעבר כזה מיותרת מכיון שהמעבר הוא בעל חשיבות רק לתהליכים עם עדיפות נמוכה. וכל התהליכים עם עדיפות גבוהה בכל מקרה יגיעו למצב *running* דרך מעבר מאחד ממצבי *ready*
- (ד) כל התשובות הקודמות הן נכונות

## שאלה 7

מה הן התכונות של האלגוריתם `guaranteed scheduling`?

- (א) האלגוריתם מבטיח שאם תהליך מגיע ראשון הוא גם ירוץ ראשון (first come first served)
- (ב) האלגוריתם מבטיח זמן שהיה (turnaround time) מזערי במערכת
- (ג) האלגוריתם מחלק את זמן החישוב (CPU) בין משתמשי מערכת (או תהליכי מערכת) במידה שווה
- (ד) כל התשובות הקודמות הן נכונות

## שאלה 8

בחרו טענה שתהיה נכונה בהנחה שהאלגוריתם האופטימאלי (optimal algorithm) הינו בר-יישום ובהנחה שכל התנאים פרט לאלגוריתם להחלפת הדפים (page replacement algorithm) זהים לגמרי:

- (א) האלגוריתם האופטימאלי מקטין את הסכנה לסחרור (thrashing) בהשוואה ל LRU
- (ב) LRU מקטין את הסכנה לסחרור (thrashing) בהשוואה לאלגוריתם האופטימאלי
- (ג) הן במקרה של LRU והן במקרה של האלגוריתם האופטימאלי רמת הסיכון להיכנס לסחרור זהה לחלוטין
- (ד) אף תשובה קודמת איננה נכונה

## שאלה 9

פונקציה `sbrk` במערכת UNIX משמשת ל:

- (א) יצירת תהליך חדש
- (ב) שינוי גודל סגמנט ה data
- (ג) יצירת קשורים סימבוליים (symbolic links)
- (ד) לקריאה/כתיבה מתוך/אל קבצים

## שאלה 10

איזה מבין האלגוריתמים הבאים להחלפת דפים אינו אלגוריתם מחסנית (stack algorithm) ?

(א) NRU (Not Recently Used)

(ב) FIFO (First In First Out)

(ג) second chance

(ד) כל האלגוריתמים ברשימה אינם מסוג אלגוריתם מחסנית

## שאלה 11

הטענות הבאות המתייחסות לניהול הזיכרון. בחרו בטענה הנכונה ביותר :

(א) שימוש ב TLB (Translation Lookaside Buffer) מקטין את מספר הגישות בזיכרון הראשי

(ב) האצה של חיפוש בטבלת הדפים המהופכת (inverted page table) מושגת ע"י השימוש ב hashing techniques

(ג) בשיטת הדפדוף (regular paging) לכל תהליך ישנה טבלת דפים משלו

(ד) כל התשובות הקודמות הן נכונות

## שאלה 12

במערכת עם ניהול זיכרון באמצעות הדפדוף (paging) נתון כי :

- כתובת מדומה (virtual address) הינה בת 32 bit

- אורך מילת הזיכרון הוא 1 byte

- גודל הדף הוא 32 Kbyte

מהו מספר הדפים המקסימאלי של הזיכרון הוירטואלי במערכת זו?

(א)  $2^{15}$

(ב)  $2^{16}$

(ג)  $2^{17}$

(ד)  $2^{18}$

## שאלה 13

לפניכם מספר טענות בנוגע לשכפול של file descriptor במערכת הפעלה UNIX. בחרו בטענה הנכונה ביותר:

- (א) במערכת הפעלה UNIX הקריאה ל select יכולה לשכפל את ה file descriptor של קובץ פתוח
- (ב) במערכת הפעלה UNIX הקריאה ל open יכולה לשכפל את ה file descriptor של קובץ פתוח
- (ג) במערכת הפעלה UNIX הקריאה ל dup יכולה לשכפל את ה file descriptor של קובץ פתוח
- (ד) אף תשובה קודמת איננה נכונה

## שאלה 14

במערכת הפעלה UNIX כל תכונות הקובץ (file attributes) למעט שם הקובץ ונתיבו שמורות ב:

- (א) I-node של הקובץ
- (ב) ברשומה של קובץ הספרייה (directory file) אשר בה נמצא הקובץ
- (ג) בטבלת הקבצים הפתוחים
- (ד) אף תשובה קודמת איננה נכונה

## שאלה 15

אחד החסרונות המובהקים של שיטת ה I-nodes היא:

- (א) ריסוק פנימי (internal fragmentation) משמעותי של שטח הדיסק
- (ב) השיטה אינה מאפשרת גישה אקראית (random access) לקובץ
- (ג) מספר גבוה של גישות לדיסק בזמן עבודה עם קבצים גדולים
- (ד) אף תשובה קודמת איננה נכונה

## שאלה 16

באיזו שכבה של תוכנת קלט/פלט (I/O software) ממומש לרוב מנגנון המטפל בשמות ארוכים של קבצים?

- (א) במערכת הטיפול בפסיקות הנוצרות ע"י ההתקנים (interrupt handling mechanism)
- (ב) בתוך תוכניות התיאום בין ההתקן לבין מערכת ההפעלה (device drivers)
- (ג) בתוך תוכנת קלט/פלט הבלתי תלויה בהתקן (device independent software)
- (ד) בשדים (daemons) לארגון הפלט ובפונקציות הספרייה ברמת המשתמש

## שאלה 17

לפניכם מספר טענות הקשורות למערכת קלט/פלט. בחרו בטענה הנכונה ביותר:

- (א) מנגנון DMA (Direct Memory Access) מקטין את מספר פסיקות הקלט/פלט (I/O interrupts) בהשוואה לשיטת interrupt-driven I/O
- (ב) אלגוריתם SSF (Shortest Seek First) לתזמון זרוע דיסק מונע הרעבה (starvation)
- (ג) מתאם ההתקן (device driver) של דיסק מורשה להשתמש בקריאת מערכת write של מערכת הפעלה על מנת לייעל את ביצועיו
- (ד) כל התשובות הקדומות הן נכונות

## שאלה 18

תהליך יחיד שנכנס ללולאה אינסופית נמצא במצב:

- (א) קיפאון (deadlock)
- (ב) הרעבה (starvation)
- (ג) סחרור (thrashing)
- (ד) אף תשובה איננה נכונה

**בהצלחה!**

## 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 5 נקודות. כל שאלה ללא תשובה מורידה רק 3 נקודות. הציון ההתחלתי הוא 100. 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**



מהו ההבדל העיקרי (מבחינת מערכת ההפעלה) בין TRAP לפסיקת החומרה ( hardware interrupt)?

- א. TRAP מעבירה את מערכת ההפעלה ממצב משתמש (user mode) למצב ראשוני (kernel mode). פסיקת החומרה מעבירה את מערכת ההפעלה ממצב ראשוני למצב משתמש.
- ב. TRAP מעבירה את מערכת ההפעלה ממצב ראשוני למצב משתמש. פסיקת החומרה מעבירה את מערכת ההפעלה ממצב משתמש למצב ראשוני.
- ג. TRAP מעבירה את מערכת ההפעלה ממצב משתמש למצב ראשוני בעקבות קריאה מתוכנתת. פסיקת מעבירה חומרה את מערכת ההפעלה ממצב משתמש למצב ראשוני בעקבות אירוע אסינכרוני כלשהו (כגון סיום ה I/O).
- ד. אף תשובה לא נכונה.

## שאלה 2

להלן פרוטוקול לפתרון של בעיית הקטע הקריטי:

process 0	process 1
<pre>flag[0] = true; while (flag[1] == true(     while (flag[1] == true(         flag[0] = false;         flag[0] = true; &lt;critical code&gt; flag[0] = false;</pre>	<pre>flag[1] = true; while (flag[0] == true(     while (flag[0] == true(         flag[1] = false;         flag[1] = true; &lt;critical code&gt; flag[1] = false;</pre>

כאשר המערך

bool flag[2];

משותף לשני תהליכים ו מאותחל ל false.

- א. הפתרון מקיים את התנאי של המניעה ההדדית
- ב. הפתרון לא תמיד מקיים את התנאי של המניעה ההדדית
- ג. הפתרון הוא פתרון סביר.
- ד. אף תשובה קודמת איננה נכונה.

## שאלה 3

לפניך פתרון אשר הוצע לבעיית הפילוסופים הסועדים :

```
#define N 5
void philposopher(int I){
    while (1) {
        think();           // philosopher is thinking
        take_fork(i);      // waits until the specified fork is available and
then seizes it
        take_fork((I+1) % N);
        eat();             // yum-yum
        put_fork(i);       // release the specified fork
        put-fork((I+1) % N);
    }
}
```

מהי הבעיה, אם בכלל, בפתרון זה?

- א. הפתרון נותן את רמת המקביליות המקסימלית האפשרית
- ב. הפתרון עלול לגרום לקיפאון (deadlock)
- ג. הפתרון לא נותן את רמת המקביליות המקסימלית האפשרית
- ד. אין בעיה בפתרון כלל וכלל

## שאלה 4

מה תהיה התחזית לריצה הבאה אם האלגוריתם לתזמון תהליכים הוא גרסה של SJF (shortest first job) המשתמש בטכניקת ה aging כשאר

• הריצות בפועל שהיו (משמאל לימין) הן 80, 40, 80, 30

•  $a=1/2$

• צפי זמן התחלתי הוא 80

א. 50

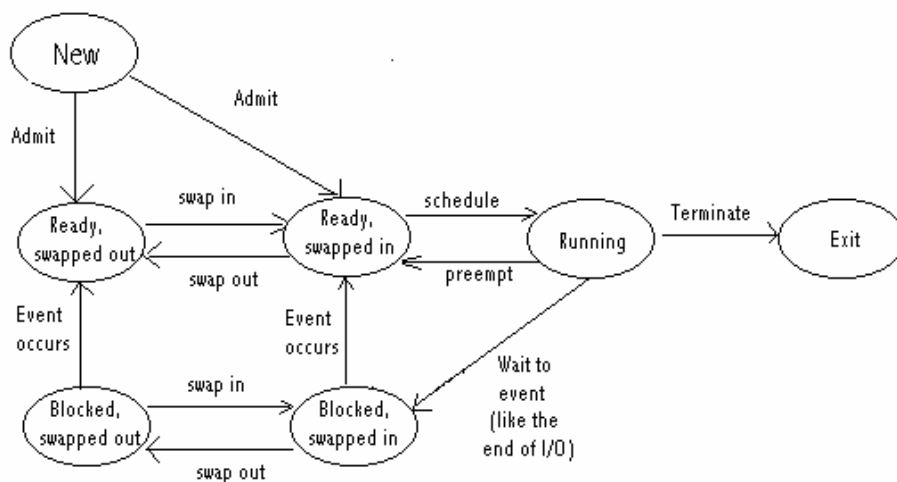
ב. 25

ג. 30

ד. 45

## שאלה 5

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



מדוע, לדעתך, צריך מעבר ממצב "Blocked, swapped out" ל "Ready, swapped out"?

א. מעבר זה נחוץ למקרה של תהליכים גדולים בלבד, אשר ממתינים לאירוע כלשהו (כגון אירוע סיום I/O)

ב. מעבר זה נחוץ למקרה של תהליכים קטנים בלבד, אשר ממתינים לאירוע כלשהו (כגון אירוע סיום I/O)

ג. מעבר זה נועד למקרה של תהליכים כשהם (בלי תלות בגודל התהליך), אשר ממתינים לאירוע כלשהו.

ד. מעבר זה איננו נחוץ כלל וכלל.

## שאלה 6

הגדלת אורך יחידת ה quantum בשיטת התזמון round robin גורמת ל:

- א. הגדלת ניצולת ה CPU
- ב. שיפור המקביליות
- ג. שיפור זמן התגובה לתהליכים אינראקטיביים
- ד. כל התשובות נכונות

## שאלה 7

זיכרון ראשי של מערכת הפעלה מסוימת הנו בגודל K 512 ומנוהל בשיטת ה variable partitions. הזיכרון פנוי בתחילת הריצה. לפניך רשימת הקצאות ושחרורי קטעי הזיכרון.

- להקצות K 70 לתהליך A.
- להקצות K 35 לתהליך B.
- להקצות K 80 לתהליך C.
- להקצות K130 לתהליך D.
- לשחרר K 70 לרגל הסיום של תהליך A.
- להקצות K 180 לתהליך E.

מהו גודלו המקסימלי של התהליך שמערכת יכולה להריץ כל עוד לא הסתיימו התהליכים B-E?

הערה: אין memory compaction.

- א. K 70
- ב. K 80
- ג. K 180
- ד. K 210

## שאלה 8

ניהול הזיכרון במערכת מסוימת מתבצע באמצעות הדפדוף (paging) כשאר האלגוריתם להחלפת הדפים הוא אלגוריתם ההזדמנות השנייה (second chance). מחרוזת ההתייחסויות (reference string) הנה (מימין לשמאל)

5, 7, 8, 2, 1, 7, 0, 5, 3, 2, 1, 0

מצב התחלתי של הזיכרון-הכל ריק. מספר המסגרות (frames) של הזיכרון הראשי הנו 4. מהו הדף המיועד להחלפה במידה ואירעה התייחסות לדף 0?

א. 2

ב. 5

ג. 7

ד. 8

## שאלה 9

מהו גודלו של מרחב הזיכרון המדומה של תהליך במערכת הפעלה עם טבלת דפים דו-שכבתית (2-level page table) כאשר

- הכתובת המדומה הנה בעלת 32 סיביות
- 10 סיביות של הכתובת המדומה הן לציון הכניסה (entry) ב top-level page table
- 10 סיביות נוספות של הכתובת המדומה הן לציון הכניסה (entry) ב second-level page table
- 12 סיביות נוספות של הכתובת המדומה הן לציון הן לציון היסט (offset)
- גודל מילת הזיכרון היא 1 byte.

א. 512 Mbyte

ב. 1 Gbyte

ג. 2 Gbyte

ד. 4 Gbyte

## שאלה 10

מערכת הקבצים של מערכת הפעלה מסוימת עושה שימוש ברשימות גישה (access controll lists – ACL). נניח כי במערכת מוגדרים 4 משתמשים Jan, Els, Jelle, Maaik אשר משתייכים לקבוצות (groups) הבאות : stuff, system, students. להלן רשימת ACL של מספר קבצים במערכת :

File0: (Jelle, \*, RWX)

File1: (Maaik, system, RWX)

File2: (Jan, \*, RW-), (Els, stuff, R--), (Maaike, \*, RW-);

File3: (\*, students, R--);

File4: (Jelle, \*, \*)

כל רשימת ACL מורכבת משלשה של uid (user ID), gid (group ID), access permissions הרשאות גישה הן RWX (read, write, execute). משמעות ה "-" במחרוזת ההרשאות היא שאין הרשאה מתאימה. משמעות ה "\*" ל gid או uid היא "כל המשתמשים" או "כל הקבוצות". איזה משתמש מנוע מקריאת קובץ מסוים מהרשימה הנ"ל, בעת שכל שאר המשתמשים ראשים לעשות זאת?

הערה: שימו לב כי file0 ניתן לקריאה/כתיבה/יצוע למשתמש Jelle ולא משנה לאיזו קבוצה הוא שייך. כל שאר המשתמשים אינם רשאים לגשת לקובץ זה!

א. Jan

ב. Jelle

ג. Maaike

ד. לא ניתן באמצעות ACL לקבוע הרשאות כאלה.

ה. אף תשובה קודמת איננה נכונה.

## שאלה 11

להלן תוצאות הבדיקה של עקביות מערכת הקבצים :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Block number
1	0	0	1	0	0	0	0	1	0	1	1	1	1	0	Blocks in use
0	0	1	0	1	1	1	0	0	1	0	0	0	0	1	Free blocks

מה אפשר להסיק לפי הנתונים הנ"ל?

- א) מערכת הקבצים הנה עקבית.
- ב) מערכת הקבצים איננה עקבית – בלוקים חסרים.
- ג) מערכת הקבצים איננה עקבית – בלוקים משוכפלים.
- ד) שתי תשובות בלבד מבין התשובות הקודמות נכונות.

## שאלה 12

התפקיד העיקרי של חוצץ במערכת הפעלה הוא :

- א. הגדלת שטח הזיכרון העומד לרשות תהליכים.
- ב. ייעול העברת נתונים בין התקנים בעלי קצב עבודה שונה.
- ג. הגדלת מהירות עבודה של התקנים איטיים.
- ד. ניצול יעיל של זיכרון.

## שאלה 13

האם תהליך יחיד שנכנס ללולאה אינסופית נמצא במצב

- א. קיפאון (deadlock)
- ב. הרעבה (starvation)
- ג. סחרור (thrashing)
- ד. אף תשובה איננה נכונה.

## שאלה 14

במערכת הפעלה UNIX לכל תהליך קיימים 2 קבצים אשר משמשים לאחסון הדפים (pages) בדיסק: קובץ ההרצה של התוכנית וקובץ מיוחד השומר העתק של חלק מהזיכרון של התהליך. לפניך אפשרויות של מיפוי החלקים השונים של התהליך לקבצים אלו. בחר את התשובה הנכונה.

- (א) data segment ממופה לקובץ המיוחד, text segment ממופה לקובץ הרצה, stack segment ממופה לקובץ הרצה.
- (ב) data segment ממופה לקובץ המיוחד, text segment ממופה לקובץ המיוחד, stack segment ממופה לקובץ הרצה.
- (ג) data segment ממופה לקובץ המיוחד, text segment ממופה לקובץ המיוחד, stack segment ממופה לקובץ המיוחד.
- (ד) data segment ממופה לקובץ המיוחד, text segment ממופה לקובץ הרצה, stack segment ממופה לקובץ המיוחד.

## שאלה 15

כידוע קריאת מערכת fork יוצרת תהליך בן עם מרחב כתובות (address space) שונה מזה של התהליך שקרא ל fork, אך זהה מבחינת התוכן. הדבר דורש העתקת תוכן מרחב הכתובות של התהליך האב ומהווה פעולה יקרה במקרה של תהליכים גדולים. זמן העתקה זה מבוזבז כמעט תמיד כי ברוב המקרים קוראים לפונקציית exec לאחר קריאת ה fork. כדי למנוע את הבזבז בזמן העתקה הוצעה קריאת מערכת חדשה vfork, אשר יוצרת תהליך בן כאשר מרחב כתובות משותף בינו לבין תהליך האב. בחר/י את המשפט הנכון:

- א. התנהגות של vfork שונה באופן מהותי מזו של fork. למשל, שינוי ערך משתנה בתהליך הבן יכול לגרום לשינוי ערך בתהליך האב.
- ב. אין הבדל משמעותי בין קריאת מערכת אחת לשנייה. הקריאה ל fork או vfork מתבצעת אך ורק למטרת הקריאה ל exec בעתיד.
- ג. אין הבדל משמעותי בין קריאת מערכת אחת לשנייה. גם לאחר הקריאה ל fork, שינוי ערך משתנה בתהליך הבן יכול לגרום לשינוי ערך בתהליך האב.
- ד. שתי תשובות בלבד בין התשובות הקודמות נכונות.

## שאלה 16



תהליך במערכת הפעלה UNIX קורא לקריאת מערכת exec. מספר תכונות של התהליך ה"ישן" עוברות בירושה אל התהליך ה"חדש" לאחר הקריאה ל system call. בחר/י תחונה אשר עוברת בירושה:

- א. signal disposition
- ב. cwd (current working directory)
- ג. שתי תשובות בלבד בין התשובות הקודמות נכונות
- ד. אף תשובה לא נכונה

## שאלה 17

היכן מוחזק מצביע למסגרת של הזיכרון הפיזי במערכת הפעלה NT כאשר מדובר בדף (page) משותף למספר תהליכים (shared page)?

- א. ב page table entry של כל אחד מהתהליכים.
- ב. ב page table entry של התהליך הראשון שעשה שימוש בדף המשותף.
- ג. ב prototype page table entry שמוצבע מ page table entry של התהליכים המשתפים את הדף.
- ד. אף תשובה קודמת איננה נכונה.

## שאלה 18

מהם בד"כ מדיניות והאלגוריתם להחלפת דפים (page replacement) של דף מדפדף (page daemon) במערכת הפעלה UNIX עם ניהול זיכרון באמצעות הדפדוף (paging)?

- א. המדיניות היא גלובלית והאלגוריתם הוא אלגוריתם השעון (clock algorithm).
- ב. המדיניות היא לוקאלית והאלגוריתם הוא אלגוריתם השעון (clock algorithm).
- ג. המדיניות היא גלובלית והאלגוריתם הוא FIFO.
- ד. המדיניות היא לוקאלית והאלגוריתם הוא FIFO.

## סוף

## 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.

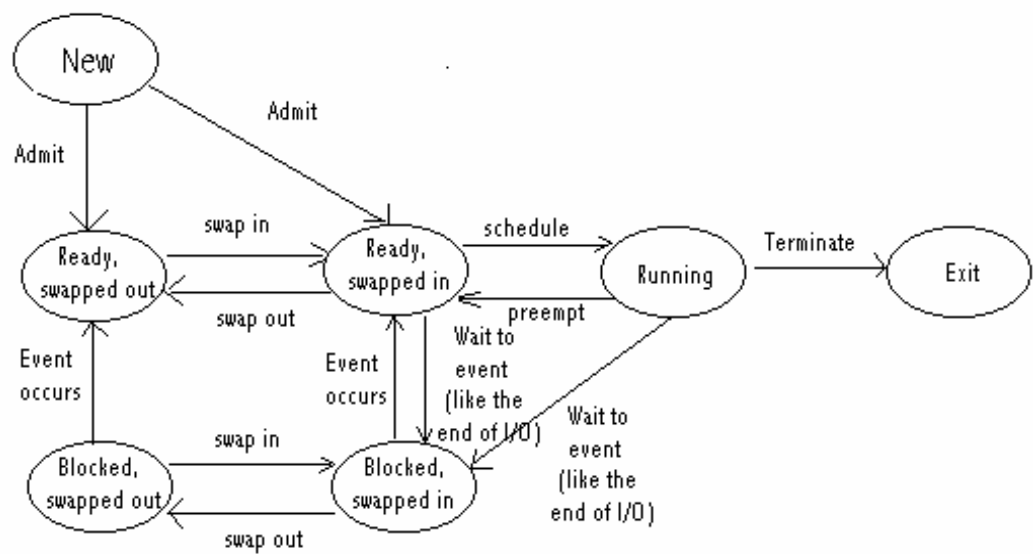
(ב) לא, כי שני תהליכים עלולים להימצא בו-זמנית בתוך קטע קריטי.

(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

## 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 וכיו')
- קובץ המורכב מרשומות (records) בגודל זהה 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 השהיה 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?

- עבור idle thread
- עבור 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 ל wait 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 (ניהולו 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. wxr bits במחרוזת הרשאות הקבצים  
ד. symbolic links

## שאלה 9

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

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

## שאלה 10

מהו interrupt vector?

- טבלה המכילה כתובות של הפונקציות האמורות להתבצע בעקבות קבלת הסיגנלים (signals) במערכת UNIX
- טבלה המכילה כתובות של device drivers לכל ה drivers שמותקנים במערכת.
- טבלה המכילה כתובות של interrupt handler routine לכל סוג של פסיקת חומרה (interrupt)
- שתיים מהתשובות נכונות

## שאלה 11

תפקידו של Paging Daemon הוא :

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

## שאלה 12

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

- שמות הקבצים ולכל קובץ מספר ה-i-node שלו
- שמות הקבצים ולכל קובץ מספר הבלוק הראשון שלו
- שמות הקבצים ולכל קובץ התכונות (Attributes) שלו
- שמות בקבצים ולכל קובץ מספר ה-i-node- והתכונות (Attributes) שלו
- שמות הקבצים ולכל קובץ מספר הבלוק הראשון והתכונות (Attributes) שלו

## שאלה 13

להלן קטע קוד של התהליך המתבצע בסביבת מערכת הפעלה UNIX :

```
Main(){
    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) ישהו פרק זמן מסוים במצב zombie?

- א. לא ייתכן
- ב. ייתכן והתהליך הוא תהליך הבן בלבד
- ג. ייתכן והתהליך הוא תהליך האב בלבד
- ד. ייתכן לגבי שני התהליכים

#### 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 בקבוצה משותפת של מסגרות בזיכרון הראשי?

- א. כן. רק בתנאי שהתהליכים מריצים את אותה תוכנית בדיוק
- ב. כן. בתנאי שכל הדפים שממופים לאותה קבוצה משותפת של מסגרות הם לקריאה בלבד
- ג. כן. רק בתנאי שהדפים שממופים באותה קבוצה משותפת של מסגרות הם דפי data
- 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 ה תהליכים	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

## 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.

hardlink במערכת הפעלה UNIX הנר:

- (א) קובץ מיחד (special file) במערכת הקבצים של מערכת ההפעלה.
- (ב) מושג המתקשר ל i-node.
- (ג) רכיב בטבלת FAT (file allocation table).
- (ד) אף תשובה קודמת אינה נכונה.

## שאלה 2.

להלן טבלה עם זמני הגעת תהליכים והזמן הדרוש לעיבוד של כל אחד מהם. חשבו כמה תהליכים מסתיימים ביחידת זמן בממוצע (throughput) כאשר אלגוריתם התזמון הוא FIFO (עיבוד באצווה - batch). מס' יחידות הזמן נמדד החל מתחילת עיבוד התהליך הראשון ועד לסיום התהליך האחרון.

תהליך	זמן הגעה	זמן CPU
A	0	1
B	1	2
C	2	1
D	3	4

(א)  $\frac{1}{2}$ .

(ב) 1.

(ג) 2.

(ד)  $\frac{3}{5}$ .

### שאלה 3.

מה התכונות של האלגוריתם הבא למניעה הדדית, עבור התהליכים  $i = 0, 1$

אתחול (שני התהליכים משתפים את המשתנה):

turn = 1

אלגוריתם:

```
while (true){
    if (turn == i) {
        // ... Critical Section
    }
    else {
        while (turn <> i) ; // wait
        // ... Critical Section
    }

    turn = 1 - i;
}
```

(א) יש מניעה הדדית ויש הגינות.

(ב) יש מניעה הדדית ואין הגינות.

(ג) אין מניעה הדדית אבל יש הגינות.

(ד) אין מניעה הדדית ובנוסף יש סכנת deadlock.

רמז: היזכרו כיצד מטפל פתרון של Peterson בהגינות לשני תהליכים.

### שאלה 4.

גרף המצבים של תהליך מכיל 3 מצבים בסיסיים: ready, blocked ו ready.

(א) המעבר מ running ל ready נובע תמיד מפעולה שבוצעה ע"י התהליך עצמו.

(ב) המעבר מ blocked ל ready נובע תמיד מפעולה שבוצעה ע"י התהליך עצמו.

(ג) המעבר מ ready ל running נובע לרוב מפעולה שבוצעה ע"י התהליך עצמו.

(ד) המעבר מ running ל blocked נובע לרוב מפעולה שבוצעה ע"י התהליך עצמו.

### שאלה 5.

אלו מבין האלגוריתמים הבאים לתזמון תהליכים סובלים מבעיית ההרעבה (starvation)?

א) FCFS (first come first served)

ב) Round-Robin ו FCFS

ג) SJF (shortest job first) ותזמון לפי עדיפות קבועה (fixed priority scheduling)

ד) SJF (shortest job first) ותזמון לפי עדיפות משתנה (priority scheduling)

### שאלה 6.

מערכת הפעלה נתונה משתמשת בשיטת ניהול הזיכרון באמצעות מחיצות בגודל משתנה (variable partitioning).

להלן גדלים של המחיצות הפנויות המקושרות ברשימה בסדר משמאל לימין :

100 K, 500 K, 200 K, 300 K, 600 K

אלגוריתם המנהל את המחיצות מתבקש לטפל בתהליכים בגודל 212 K, 417 K, 112 K, 426 K (סדר ההגעה הוא משמאל לימין).

איזו שיטה להקצאה במחיצות תהיה העדיפה מבין 2 השיטות הבאות: best-fit או first-fit בנתונים הנ"ל?

א) first-fit עדיפה על best-fit.

ב) best-fit עדיפה על first-fit.

ג) אין הבדל ממשי בניהם.

ד) לא ניתן לספק תשובה עקב חוסר הנתונים.

### שאלה 7.

מספר הסיביות (bits) של כתובת מדומה (virtual address) להיסט (offset) הנו 12. מילת זיכרון הנה באורך 32 סיביות. מהי גודלה של מסגרת (frame) בזיכרון הפיזי?

(א) 16K.

(ב) 8K.

(ג) 128K.

(ד) 4K.

### שאלה 8.

להלן רשימת אלגוריתמים להחלפת דפים בזיכרון מדופדף: (1) LRU, (2) FIFO, (3) האלגוריתם האופטימלי, (4) אלגוריתם ההזדמנות השניה (second chance). אלו מן האלגוריתמים הללו סובלים מאנומלית בלאדי (Belady)?

(א) FIFO ו LRU.

(ב) FIFO.

(ג) LRU ואופטימלי.

(ד) FIFO ואלג' ההזדמנות השניה.

### שאלה 9.

רשימות היכולות (capability lists) מוחזקות בד"כ במרחב הזיכרון של התהליך. כיצד מובטח שהמשתמש לא ישנה את תוכן הרשימה?

(א) ע"י שימוש בתכונת ה tagged architecture של החומרה.

(ב) ע"י הצפנת הרשימה במפתח סודי שלא ידוע למשתמש.

(ג) ע"י כך שכדי לשנות את רשימת היכולות נדרשות הרשאות מנהל מערכת.

(ד) 2 מן התשובות הקודמות נכונות.

(ה) שלושת התשובות הראשונות נכונות.

### שאלה 10.

האם קיימת דרך סבירה להילחם בתופעת הדלפת מידע ע"י שימוש בערוצים חסויים (covert channels) בצורה מקיפה?

- (א) כן, ע"י הקצאה ושחרור אקראי של משאבים.
- (ב) כן, באמצעות חסימת האפשרות ליצירה ומחיקה של קבצים ע"י תהליכים החשודים להדלפה.
- (ג) לא.
- (ד) כן, ע"י צירוף של השיטות שהוזכרו בסעיפים א' ו ב'.

### שאלה 11.

דיסקים מסוימים מחזיקים סוג של זיכרון מטמון שנקרא `track-at-a-time cache`. בזיכרון זה נשמרים כל הבלוקים שהראש הקורא עבר עליהם במסלול עד שהגיע לסקטור המבוקש. זיכרון זה שקוף ל `device independent software` (כלומר על קיום זיכרון זה ברמת התוכנה ידוע רק ל `device driver`). מה הם החסרונות של זיכרון מטמון מסוג זה?

- (א) סיבוכיות נוספת של `device driver`, חומרה מתוחכמת יותר, בזבוז CPU על העתקת בלוקים מן הזיכרון המטמון אל הזיכרון הראשי.
- (ב) סיבוכיות נוספת של `device driver`, חומרה מתוחכמת יותר, מספר רב יותר של תזוזות זרוע הדיסק.
- (ג) סיבוכיות נוספת של `device driver`, חומרה מתוחכמת יותר, זמן המתנה רב יותר לסיבוב הדיסק עד שראש הקורא נעמד על הסקטור הדרוש.
- (ד) אף תשובה קודמת אינה נכונה.

### שאלה 12.

לשם מה מתעורר הצורך בנעילת (locking) דפים בזיכרון הפיזי?



- (א) למנוע מצב של הוצאת דף אליו מתבצע DMA transfer (וכתוצאה מכך למנוע שחלק המידע מהדיסק ייכתב על הדף החדש שנטען במקום זה שיוצא).
- (ב) כדי לממש מנגנון הבטחה (protection) מפני תהליכים שמנסים לשכתב קטע קוד (code segment) של עצמם.
- (ג) בדרך זו ניתן להבטיח כי כל הדפים מקבוצת העבודה (working set) של התהליך יישארו בזיכרון הפיזי.
- (ד) 2 מן התשובות הקודמות נכונות.

### שאלה 13.

- ☐ במערכת מסוימת 4 משאבים מאותו סוג ושלושה תהליכים.
  - ☐ כל תהליך יכול לדרוש 2 משאבים לכל היותר.
  - ☐ 2 תהליכים אינם יכולים להשתמש במשאב בו זמנית.
  - ☐ תהליך אינו משחרר משאבים שהוקצו לו עד שיסתיים.
  - ☐ אין חילץ כפוי של משאבים.
  - ☐ תהליך שמקבל כל המשאבים הדרושים מסתיים בזמן סופי.
- האם יש צורך בטיפול בקיפאון במערכת?

- (א) אין צורך בכך מפני שאין חשש להיווצרות הקיפאון.
- (ב) יש צורך בכך מפני שעלולה להיווצר המתנה מעגלית.
- (ג) יש צורך בכך מפני שעלול להיווצר קיפאון, אומנם לא נוצרת המתנה מעגלית.
- (ד) אף תשובה קודמת אינה נכונה.

### שאלה 14.

להלן מדיניות הקצאת משאבים במערכת מסוימת:

תהליך יכול לבקש/לשחרר משאבים בכל עת. לכל תהליך קיים וקטור של משאבים אשר להם ממתין תהליך. אם לא ניתן לספק בקשה למשאב כלשהו, אז המערכת עוברת על תהליכים אשר נמצאים במצב (blocked) עקב המתנה למשאבים. אם הם מחזיקים במשאב הדרוש, אז המשאב נחלץ מידיהם וניתן לתהליך הדורש. וקטור המשאבים אשר להם ממתין התהליך מתעדכן בהתאם. האם יכול להיווצר קיפאון במערכת כזאת?

#### דוגמא:

למשל במערכת עם 3 סוגי משאבים וקטור סך המשאבים במערכת הוא (4,2,2). תהליך P0 מבקש (2,2,1), P1 מבקש (1,0,1) ושניהם מקבלים את בקשתם. אם לאחר מכן P0 מבקש (0,0,1) הוא עובר למצב blocked (אין משאב 3 פנוי). כעת אם P2 מבקש (2,0,0) הוא מקבל משאב אחד שפנוי ואחד נלקח מ P0 (כי P0 נמצא במצב blocked). וקטור המשאבים שהוקצו ל P0 מתעדכן ל (1,2,1) ווקטור המשאבים המבוקשים מתעדכן ל (1,0,1).

- (א) כן, כי עלולה להיווצר המתנה מעגלית בין התהליכים.
- (ב) כן, כי יש חילוף משאבים.
- (ג) לא, כי לא מתקיים תנאי החזק והמתן (hold and wait).
- (ד) לא, אך עלול לקרות שתהליכים ישהו זמן לא מוגבל במצב blocked.

#### **שאלה 15.**

ניתן לייעל את קריאת המערכת fork במערכת UNIX :

- (א) ע"י שיתוף קטעי זיכרון ושכפולם לפי הצורך.
- (ב) ע"י ביצוע קריאה לקריאת מערכת exec מייד לאחר חזרה מ fork.
- (ג) ע"י השהיית תהליך הבן לזמן מה כדי להבטיח שתהליך האב יתוזמן קודם.
- (ד) כל התשובות נכונות.

#### **שאלה 16.**

קריאת המערכת readlink במערכת UNIX נועדה ל:

- א) קריאת התוכן של קובץ הספרייה.
- ב) קריאת התוכן של קובץ ה soft link.
- ג) קריאת התוכן של קובץ המוצבע ע"י soft link.
- ד) שליפת מידע על קובץ ה softlink.
- ה) שליפת מידע על קובץ המוצבע ע"י softlink.

#### שאלה 17.

האם קיימים במערכת הפעלה NT מנגנונים להרצת קבצי הרצה (בינאריים) של MS-DOS ?

- א) כן, עבור גרסאות NT למעבדים אשר אינם תומכים ב instruction set של מעבדי Intel, קיים מנגנון האמוליציה (emulation) של פקודות ה- instruction set.
- ב) כן, קיים מנגנון הטיפול בקריאות המערכת MS-DOS דרך ה environmental subsystem.
- ג) כן, אך לא מובטחת תאימות בינארית (binary compatibility) מלאה.
- ד) כל התשובות הקודמות נכונות.
- ה) 2 תשובות קודמות בלבד נכונות.

הערה: אמוליצית פקודות ה- instruction set פירושה תרגום בין פקודות ה- instruction set של מעבד אחד לפקודות ה- instruction set של מעבד אחר.

#### שאלה 18.

מה הוא ה idle thread במערכת הפעלה NT ?

- א) ה thread עם עדיפות הכי גבוהה במערכת.
- ב) ה thread אשר נמצא במצב waiting.
- ג) ה thread אשר רץ כל עוד אין threads במצב ready.
- ד) ה thread אשר הסתיים.

בהצלחה

## 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

## **בחינה במערכות 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 את אות התשובה בעיגול, רק על גבי השאלון עצמו. אין צורך לסמן את התשובה בשום מקום אחר.
- ב.** הציון נקבע על פי מספר השגיאות : כל שגיאה מורידה 5 נקודות.  
כל שאלה ללא תשובה מורידה רק 4 נקודות.
- ג.** כל חומר עזר אסור בשימוש בזמן הבחינה.
- ד.** במקרה של ספק בהבנת ניסוח השאלה ניתן לכתוב ליד השאלה, רק על גבי השאלון עצמו, הערה המבהירה את דרך הבנתך את השאלה. אין לכתוב נימוקים לתשובה!
- ה.** משך הבחינה 3 שעות.

**בהצלחה!**

### שאלה 1

במערכת DOS ההתחלה של FAT נראית כך :

0	1	2	3	4	5	6	7	8	9	10
-1	3	9	7	10	-1	-1	2	-1	5	6

ידוע שהבלוק הראשון של קובץ מסוים הוא בלוק 1.  
כמה בלוקים יש בקובץ הזה ?

- א. 5.
- ב. 2.
- ג. 6.
- ד. 4.

### שאלה 2

מתאמי התקנים (Device Drivers):

- א. מונעים תלות בהתקן פיסי ומאפשרים ממשק אחיד.
- ב. מבצעים Spooling של עבודות קלט/פלט.
- ג. ממפים את שמות התקנים להתקנים פיסיים המתאימים.
- ד. מבטיחים גודל בלוק אחיד לכל התקנים.
- ה. שתיים מהתשובות נכונות.
- ו. שלוש מהתשובות נכונות.

### שאלה 3

לדיסק הגיעו 5 בקשות לקריאת בלוקים במסלולים (tracks): 11, 22, 15, 18, 3 בסדר מימין לשמאל, כאשר הראש הקורא נמצא במסלול 17- והוא זז בכיוון המסלולים המרוחקים יותר. (גדולים מ-17). מהו האלגוריתם האופטימלי לחיפוש בדיסק במקרה הזה ?

- א. FCFS. First Come - First Served
- ב. SSF. Shortest Seek Time First
- ג. Elevator. מעלית.
- ד. SCAN. סריקה. (עליה כמו באלגוריתם המעלית וירידה רצופה ללא עצירות).

### שאלה 4

האם תוכניות TSR (Terminate & Stay Resident) קיימות ב-UNIX ו/או ב-DOS ?

- א. רק ב-UNIX, כי UNIX מאפשרת ריבוי תהליכים ו-אילו DOS לא.
- ב. רק ב-UNIX, כי ב-UNIX שיטת ניהול זיכרון מתאימה לריבוי תהליכים.
- ג. רק ב-DOS, כי ב-DOS משתמשים ב-FAT לניהול מערכת קבצים וב-UNIX לא.
- ד. רק ב-DOS, כי ב-DOS משתמשים יכולים להגדיר פסיקות וב-UNIX לא.
- ה. גם ב-UNIX וגם ב-DOS.

### שאלה 5

זכרון המטמון (Buffer Cache) עוזר:

- א. לחסוך מקום בדיסק.
- ב. לחסוך זיכרון פיסי.
- ג. לחסוך בהעברת קבצים מדיסק לזיכרון ולהפך.
- ד. לייעל את עבודתו של מנגנון החלפת דפים בזיכרון המדופדף.

## שאלה 6

הפתרון הבא הוצע לבעיית הקטע הקריטי עבור שני תהליכים  $P_i$  ו- $P_j$ .  
להלן הפרוטוקול עבור התהליך  $P_i$ :

```
...
while (lock == TRUE);
lock = TRUE;
if (turn == j) {
    turn = i;
    flag[i] = TRUE;
}

... /* Critical Section */

turn = j;
flag[i] = FALSE;
lock = FALSE;
...
```

האם זהו פתרון סביר ?

- א. כן.
- ב. לא, כי שני תהליכים עלולים להימצא בו-זמנית בתוך קטע קריטי.
- ג. לא, כי הפתרון הזה עלול לגרום לקיפאון.
- ד. לא, כי תהליך מחוץ לקטע קריטי עלול למנוע מתהליך אחר להיכנס לתוך קטע קריטי.
- ה. לא ניתן לענות לפי הנתון.

## שאלה 7

באיזו שיטה מטפלים בבעיית הקיפאון במערכת UNIX ?

- א. בניית גרף הקצאת משאבים.
- ב. גילוי והתאוששות (Detection & Recovery).
- ג. התחמקות (Avoidance).
- ד. מניעה (Prevention).
- ה. אף תשובה אינה נכונה.

## שאלה 8

נתון : ישנם שלושה תהליכים A, B ו-C. לכל אחד מהתהליכים יש עדיפות שונה. ידוע כי עדיפותו של תהליך A היא הנמוכה ועדיפותו של תהליך C היא הגבוהה.  
זמן הריצה הרצוף של תהליך הוא : Quantum=1  
ניתן להניח כי זמן החלפת תהליכים (Context Switch) הוא זניח.

להלן נתוני התהליכים :

תהליך	זמן הגעה	עדיפות	זמן-ביצוע
A	0	1	10
B	5	2	4
C	8	3	3

מהו זמן העבוד הממוצע של תהליך כאשר מנגנון התזמון הוא תזמון עם עדיפויות ?

- א. 17
- ב. 13.33
- ג. 5.67
- ד. 9
- ה. 11

## שאלה 9

מהן המגבלות של קישור קבצים קשיח - Hardlink?

- א. לא ניתן ליצור hardlink לספרייה.
- ב. לא ניתן ליצור hardlink לקובץ במתקן אחר.
- ג. Hardlink מבזבז מקום בדיסק.
- ד. אינו מאפשר גישה אקראית לקובץ.
- ה. אף תשובה אינה נכונה.

## שאלה 10

מדוע משתמשים ב-open במקום לתת נתיב (Full Path) מלא ל-read ול-write?

- א. על מנת לפתור את בעיית הקיפאון כאשר משתמשים בקבצי נעילה.
- ב. כדי לחסוך זמן בפענוח הנתיב ומציאת קובץ.
- ג. על מנת ליעל שימוש בקבצים במשותף ע"י מספר תהליכים.
- ד. שתיים מהתשובות נכונות.
- ה. אף תשובה אינה נכונה.

## שאלה 11

מחרוזת מרחק (Distance String) מכילה את המידע הבא:

- א. סדרת מרחקים בין מחיצות בחלוקת זיכרון למחיצות בגודל משתנה.
- ב. מידע על הבלוקים הפנויים בדיסק.
- ג. סדרת מספרי דפים לפי סדר הגישה.
- ד. אף תשובה אינה נכונה.
- ה. שתיים מהתשובות נכונות.

## שאלה 12

תפקידו של שד הדפדוף (Paging Daemon) הוא:

- א. לדאוג להחלפת דפים של תהליכים העוסקים בפעולות קלט/פלט.
- ב. לשמור על מספר מינימלי מוגדר של Page Frames פנויים.
- ג. לשמור סטטיסטיקות ומידע נחוץ עבור מנגנון החלפת דפים.
- ד. שתיים מהתשובות נכונות.
- ה. תשובות (א), (ב), (ג) נכונות.

## שאלה 13

נתונה תוכנית בשם "chd" הכתובה בשפת C בסביבת UNIX.

```
void main(int argc, char *argv[])
{
    chdir(argv[1]);
}
```

משתמש בשם משה מריץ תוכנית זו בספריית הבית שלו "/home/moshe" וזאת, ע"י פקודת: "chd ..". באיזה ספרייה ימצא משה כאשר התוכנית תסיים את עבודתה?

- א. /home
- ב. /
- ג. /home/moshe
- ד. /home במידה וקריאת מערכת chdir הצליחה ו- /home/moshe במידה והפקודה נכשלה.

#### שאלה 14

האם פתרון התור (Strict Alternation) מונע בוודאות כניסה של שני תהליכים שונים לקטע קריטי משותף?

- א. כן, בהחלט.
- ב. לא, כי הפתרון אינו מקיים את אחד מהתנאים ההכרחיים לפתרון בעיית הקטע הקריטי.
- ג. לא, כי הפתרון מהווה קטע קריטי בפני עצמו.
- ד. אף תשובה קודמת אינה נכונה.

#### שאלה 15

מה ההבדל בין קריאת מערכת לפונקצית ספריה ?

- א. קריאת מערכת יכולה להיות ממומשת בשפת מכונה בלבד.
- ב. קריאת מערכת מתבצעת במצב גרעין.
- ג. פונקצית ספריה הינה פעולה אטומית.
- ד. אף תשובה קודמת אינה נכונה.

#### שאלה 16

במערכת UNIX בכל i-node ישנן 10 כתובות דיסק עבור בלוקים רגילים ובנוסף כתובת אחת לכל אחד מהבלוקים triple indirect, double indirect, single indirect. כל כתובת דיסק היא בת 32 בייט, גודל בלוק – 0.5K. כמה סה"כ בלוקים (מלבד ה- i-node עצמו) נצטרך להקצות על מנת לשמור קובץ בגודל 115K ?

- א. 490.
- ב. 128.
- ג. 256.
- ד. 245.
- ה. 123.

#### שאלה 17

באיזו שכבה של תוכנת קלט/פלט הופכים מספרים מתצוגה בינארית לקודי ASCII ?

- א. תוכנה ברמת משתמש.
- ב. תוכנה בלתי תלויה בהתקן.
- ג. מתאם התקן (driver).
- ד. Interrupt Handler.

#### שאלה 18

אילו מהפעולות הבאות מועילות ליישום מודל קבוצת העבודה ?

- א. הבאת דפים מראש (Prepaging).
- ב. הבאת דפים לפי דרישה (Paging on Demand).
- ג. החלפת דפים מקומית (local).
- ד. שתיים מהתשובות נכונות.
- ה. אף תשובה אינה נכונה.

סוף



## שאלה 1

מה משותף בין תהליך אב לתהליך בן מייד לאחר הצלחתה של קריאת מערכת `fork` במערכת הפעלה UNIX?

- (א) סגמנט קוד (code segment)
- (ב) מפת אופן הטיפול בסיגנלים (signal disposition map)
- (ג) קבצים פתוחים (במידה והיו כאלה)
- (ד) כל התשובות הקודמות הן נכונות

## שאלה 2

מהו הפלט שאיננו יכול להיווצר כתוצאה מההרצה של התוכנית הבאה?

```
int is_terminated = 0;

void sig_handler(int signo){
    printf("3");
}

main(){
    int status;
    signal(SIGCHLD, sig_handler);
    if (fork() == 0){
        is_terminated = 1;
        exit(0);
    }
    if (is_terminated != 1){
        pause(); // waits for a signal
        printf("1");
    }
    wait(&status);
    printf("2");
}
```

- הערות: (1) ניתן להניח כי כל קריאות המערכת "מצליחות"
- (2) ניתן להניח כי הדפסת תו בודד היא פעולה אטומית
- (3) יש לקרוא את הפלטים משמאל לימין

(א) 312.

(ב) 32.

(ג) 3.

(ד) אף תשובה קודמת איננה אפשרית

### שאלה 3

בחרו את הטענה הנכונה :

- (א) פעולת החיפוש בזיכרון מטמון (cache) מתבצעת ע"י מעבר סדרתי על השורות של הזיכרון
- (ב) טבלת דפים מהופכת (inverted page table) תמיד תהיה גדולה יותר מן הטבלה הרגילה
- (ג) פענוח מענים הוא מעשי גם ללא תמיכתה של חומרה ע"י סימולציה של הפענוח בתוכנה
- (ד) אף תשובה קודמת איננה נכונה

### שאלה 4

מניעת הקיפאון (deadlock) ע"י תקיפת התנאי "מניעה הדדית" באמצעות שיתוף משאבים איננה ישמה בכל המקרים כי :

- (א) לא כל משאב ניתן לשיתוף
- (ב) גורמת לכניסה למצב לא בטוח (unsafe state)
- (ג) גורמת לכניסה למצב בטוח (safe state)
- (ד) אף תשובה קודמת איננה נכונה

### שאלה 5

לפניכם רשימת מדדים של אלגוריתמים לתזמון תהליכים. בחרו מתוך הרשימה מדד שבדרך כלל מתאים לתהליכי זמן אמת (real-time), אך איננו מתאים בדרך כלל לתהליכי batch.

- (א) ניצול CPU
- (ב) הגינות (fairness)
- (ג) עמידה בזמנים (meeting deadlines)
- (ד) כל התשובות הקודמות הן נכונות.

## שאלה 6

לפניכם פתרון של בעיית יצרן-צרכן באמצעות מבנה פיקוח (monitor). בחרו באיזה מקרה/באלו מקרים יהיה הפתרון עדיין נכון כאשר ההנחה היא שבעקבות signal כל התהליכים אשר היו במצב חסום על condition variable הופכים להיות ready to run.

```
monitor ProducerConsumer
condition full, empty;
integer count;

procedure insert(item: integer)
begin
    if count = N then wait(full);
    insert_item(item);
    count := count + 1;
    if count = 1 then signal(empty);
end

function remove :integer
begin
    if count = 0 then wait(empty);
    remove = remove_item();
    count = count - 1;
    if count = N-1 then signal(full);
end;

count = 0;
end monitor;

procedure producer
begin
    while true do
    begin
        item = produce_item;
        ProducerConsumer.insert(item);
    end;
end

procedure consumer
begin
    while true do
    begin
        item = ProducerConsumer.remove;
        consume_item(item);
    end
end
```

(א) יצרן – צרכנים

(ב) יצרנים-צרכן

(ג) יצרנים-צרכנים

(ד) כל התשובות הקודמות הן נכונות

## שאלה 7

תכונת הלוקאליות של תוכניות משפיעה על המערכת באופן הבא :

- (א) מקטינה את זמן החישוב (user time) של התוכנית מכיוון שהשימוש ב cache של ה CPU יהיה יותר יעיל.
- (ב) מקטינה את הגודל של ה working set
- (ג) מקטינה את כמות ה page faults
- (ד) כל התשובות הקודמות הן נכונות

## שאלה 8

כאשר קצב ה page faults במערכת גבוה, מערכת ההפעלה נוקטת בפעולה הבאה:

- (א) מחליפה את האלגוריתם להוצאת דפים (page replacement algorithm) באלגוריתם אחר
- (ב) מבצעת swap in של תהליך (תהליכים) להגברת קצב ה page faults
- (ג) מקטנה את כמות הזיכרון המוקצה לשורות של TLB (translation lookaside buffer)
- (ד) אף תשובה קודמת איננה נכונה

## שאלה 9

הדרישה לבקש משאבים בסדר עולה מונוטוני לפי מספור המשאבים הקבוע מראש יכולה לגרום לתופעה הבאה :

- (א) דשדוש (thrashing) של התהליכים הכפופים לדרישה
- (ב) המתנה מעגלית בין התהליכים הכפופים לדרישה
- (ג) הקטנת מספר פסיקות הדף שגורמים תהליכים הכפופים לדרישה
- (ד) אף תשובה קודמת איננה נכונה

## שאלה 10

נסמן ב- $p$  את ההסתברות לכך שתהליך מחכה לסיומה של פעולת I/O וב- $n$  נסמן את מספר התהליכים בזיכרון. כדי להגיע לנוסחה  $1-p^n$  המתארת את ניצול ה-CPU יש להניח הנחה (הנחות) הבא (הבאות):

- (א)  $n$  תהליכים הם בלתי תלויים זה בזה
- (ב)  $n$  תהליכים הם תלויים זה בזה
- (ג) אלגוריתם התזמון הוא אלגוריתם ה-RR (round-robin)
- (ד) אלגוריתם התזמון הוא האלגוריתם המבוסס על עדיפויות משתנות
- (ה) שתי תשובות מבין התשובות הקודמות הן נכונות

## שאלה 11

במערכת עם ניהול זיכרון באמצעות הדפדוף (paging) נתון כי:

- כתובת מדומה (virtual address) הנה בת 16 סיביות (bits).
- גודל מילת זיכרון הוא 1 byte.
- גודל טבלת דפים רגילה (page table) הוא 0.5 Kbytes.
- גודלה של כניסת טבלת הדפדוף (page table entry) הוא 4 bytes.

מהו גודל הדף?

- (א) 128 bytes
- (ב) 256 bytes
- (ג) 512 bytes
- (ד) 1 Kbyte

## שאלה 12

פונקציה exec במערכת UNIX משמשת ל:

- (א) טעינת תהליך חדש
- (ב) שינוי גודל סגמנט ה data
- (ג) יצירת קשורים סימבוליים (symbolic links)
- (ד) לקריאה/כתיבה מתוך/אל קבצים

## שאלה 13

בחרו מן הרשימה הבאה אלגוריתם להחלפת דפים (page replacement algorithm) שמהווה את הקירוב הטוב ביותר לאלגוריתם האופטימאלי ברוב המקרים.

(א) NRU (Not Recently Used)

(ב) Second Chance

(ג) WSClock

(ד) כל האלגוריתמים הנ"ל טובים במידה זהה

## שאלה 14

נתונה מחרוזת התייחסויות (reference string) הבאה :

0 2 1 3 4 3 2 1

מספר הדפים שהוקצו לתהליך הוא 3. בהינתן שאלגוריתם להחלפת הדפים הוא FIFO, מהו

מספר פסיקות דף (page faults) שמתרחש כאשר מדובר ב demand paging?

(א) 4

(ב) 5

(ג) 6

(ד) 7

## שאלה 15

הראש הקורא של דיסק נמצא על מסלול (track) מספר 53 (זהו מצב התחלתי). בקשות שממתינות

לביצוע מתייחסות למסלולים 67, 65, 124, 14, 122, 37, 183, 98. (משמאל לימין). באיזה

אלגוריתם לתזמון זרוע הדיסק הייתם בוחרים למקרה ספציפי זה, על מנת להקטין את תזוזת

הראש הקורא?

(א) FCFS (first come first served)

(ב) אלגוריתם המעלית - elevator algorithm (במצב התחלתי כוון התנועה של הראש הקורא הוא

לכוון המסלולים שמספרם גדול יותר)

(ג) SSF (shortest seek first)

(ד) לא ניתן לקבוע את התשובה על סמך נתוני השאלה

## שאלה 16

סגמנט ה-data של תהליך במערכת הפעלה UNIX מכיל :

- (א) את הוראות המכונה של התוכנית המתבצעת
- (ב) את המשתנים הגלובאליים של התוכנית
- (ג) את המשתנים האוטומאטיים (automatic variables)
- (ד) אף תשובה קודמת איננה נכונה

## שאלה 17

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

- גודל הבלוק במערכת הקבצים הוא 2 Kbyte
- כתובת הבלוק היא 8 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

גודלו של קובץ מסוים במערכת 632 Kbyte. מהי כמות הבלוקים שדרושה להחזקת קובץ זה במערכת הקבצים (לא כולל את הבלוק שמכיל את ה i-node של הקובץ)?

(א) 316

(ב) 317

(ג) 318

(ד) 319

(ה) 320

## שאלה 18

כאשר מדובר על מבנה מערכת הפעלה לפי מודל שרת-לקוח (client-server model) מהי התכונה אשר מהווה חסרון מובהק של המודל?

- (א) העדר מבנה כלשהו. המערכת היא אוסף שגרות אשר כל אחת מהן יכולה לקרוא לשגרה אחרת מן האוסף
- (ב) חוסר אפשרות התאמה למערכות מבוזרות (distributed systems)
- (ג) תקורת (overhead) התקשורת בין רכיבי המערכת
- (ד) כל התשובות הקודמות הן נכונות

## שאלה 19

במערכת קבצים FAT 12 הסקטורים הם בעלי 512 bytes. טבלת FAT תופסת 9 סקטורים בדיסק כאשר שתי השורות (entries) הראשונות של הטבלה הן שמורות. מהו גודלו המקסימאלי של קובץ במערכת הקבצים FAT 12 באופן תיאורטי (כלומר ללא אילוצי חומרה)?

- (א) 3070
- (ב) 3072
- (ג) 3074
- (ד) 3076

## שאלה 20

כדי לספק אפשרות של סנכרון תהליכים באמצעות מבנה פיקוח (monitor) ניתן להגדיר משתנים מיוחדים הנקראים condition variables. Signal(cond var) ו Wait(cond var) הן שתי פונקציות המוגדרות על condition variables. מימוש של מבני הפיקוח שהוצע ע"י Hansen דורש כי הקריאה ל Signal תיעשה כפעולה אחרונה של פרוצדורת מבנה פיקוח (monitor procedure). מדוע?

- (א) דרישה זו מסייעת למנוע מצב של המתנת מספר תהליכים על condition variable
- (ב) דרישה זו נחוצה כדי שעל condition variable יוכלו להמתין מספר תהליכים
- (ג) דרישה זו מבטיחה ששני תהליכים לא ימצאו בו זמנית במבנה פיקוח אחד
- (ד) כל התשובות הקודמות נכונות

בהצלחה!