

תכנות מתקדם - שאלות מבחן לתרגול

(תשובות בעמוד האחרון)

מבחן מספר 1

1. ה-LRU הוא:

- a. שיטת המרה אקסה-דצימלית.
- b. מדיניות החלפת מטמון - תפקידו לזרוק את הקריאות הכי נפוצות.
- c. מדיניות החלפת מטמון - תפקידו לזרוק את הקריאות הכי פחות נפוצות.
- d. שיטה למציאת דפים בטבלת הדפים באמצעות שארית חלוקה של כתובות בינאריות.

2. מה תדפיס התוכנית?

```
char buf1[] = "AAAAAAAAAAAAAAAAAAAA";
char buf2[] = "BBBBB";

int main(void) {
    int fd, pos1, pos2, pos3, pos4;
    fd = open("out.txt", O_WRONLY | O_APPEND | O_CREAT, S_IRUSR | S_IWUSR);
    write(fd, buf1, 20);
    pos1 = lseek(fd, 0, SEEK_CUR);
    write(fd, buf2, 5);
    pos2 = lseek(fd, 0, SEEK_END);
    pos3 = lseek(fd, 0, SEEK_SET);
    write(fd, buf1, 20);
    pos4 = lseek(fd, 100, SEEK_END);
    printf("pos1= %d , pos2= %d , pos3= %d , pos4= %d \n", pos1, pos2, pos3, pos4);
}
```

- a. pos1= 20 , pos2= 25 , pos3= 0 , pos4= 145
- b. pos1= 20 , pos2= 5 , pos3= 25 , pos4= 125
- c. pos1= 20 , pos2= 5 , pos3= 0 , pos4= 125
- d. pos1= 20 , pos2= 25 , pos3= 0 , pos4= 100

3. ניהול הבלוקים הפנויים. למדנו שניתן לסמן את הבלוקים הפנויים בזיכרון בעזרת מערך של ביטים (BitMap). כאשר כל ביט מייצג לנו בלוק בדיסק. אם גודל הדיסק הוא 1TB וכל בלוק הוא בגודל 4kB, מה יהיה גודל מערך הביטים BitMap?

- a. גודל המערך יתפוס 16MB
- b. גודל המערך יתפוס 4kB
- c. גודל המערך יתפוס 32MB
- d. גודל המערך יתפוס 64MB

4. איך אפשר למנוע מהתהליך הקורא לעבור למצב שינה כאשר ה-pipe ריק וקיים תהליך שסיים לכתוב?

- a. הקורא צריך לסגור את צד הקריאה והכותב צריך לסגור את צד הכתיבה.
- b. הקורא והכותב צריכים לסגור את צד הכתיבה.
- c. הקורא והכותב צריכים לסגור את צד הקריאה.
- d. כל התשובות נכונות.

5. תרגום כתובת במפת דפים היררכית. נניח כתובת של 32bit, בטבלת דפים עם שתי רמות, נחלק את הכתובת לשלושה חלקים: p_1, p_2, d . נתונה הכתובת 0x00403004. חשבו את האינדקסים p_1, p_2 ואת ה-offset d .

- a. $p_1 = 0, p_2 = 40, d = 3004$
- b. $p_1 = 1, p_2 = 3, d = 4$
- c. $p_1 = 0, p_2 = 4, d = 4$
- d. $p_1 = 4, p_2 = 3, d = 4$

6. החלפת דף OPT. נתונים 3 דפים פיסיים וסדרת מספרי הדפים:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

מה מספר הפסיקות לפי אלגוריתם OPT (כולל פסיקות הכרחיות)?

- a. 6 פסיקות
- b. 9 פסיקות
- c. 7 פסיקות
- d. 3 פסיקות

7. עם איזה שיטה malloc תנהל את הקצאות הבלוקים הפנויים בזמן הטוב ביותר (זמן קבוע)?

- a. ללא מצביעים.
- b. באמצעות רשימות.
- c. באמצעות מצביעים.
- d. שיטת Best fit.

8. האם התרחיש הבא אפשרי (בחרו את התשובה הנכונה ביותר):

$TLB = miss, \quad Page\ Table = hit, \quad Cache = hit$

- a. אפשרי, לא כל דף חייב להיות ב-TLB אבל הוא בהכרח חייב להיות בטבלת הדפים. במידה והנתון היה לאחרונה בשימוש אז הוא גם נמצא במטמון.
- b. לא אפשרי, כל דף שנמצא ב-Page table הוא גם נמצא ב-TLB כי הוא מוכל בתוכו.
- c. לא אפשרי, הנתון לא יכול להתקיים במטמון אם הדף לא קיים בזיכרון.
- d. לא אפשרי, שאר התשובות המוצגות לא מדויקות.

9. פסיקות (Interrupts)

- a.** חלוקה באפס תגרום לפסיקה פנימית, פסיקת שעון תגרום לפסיקה חיצונית.
- b.** פסיקת שעון תגרום לפסיקה פנימית, חלוקה באפס תגרום לפסיקה חיצונית.
- c.** ניסיון לבצע פקודה לא חוקית יגרום לפסיקה חיצונית, חלוקה באפס תגרום לפסיקה פנימית.
- d.** ניסיון לבצע פקודה לא חוקית יגרום לפסיקה פנימית, חלוקה באפס תגרום לפסיקה חיצונית.

10. קריאת מערכת (System Call)

- a.** קריאת מערכת תגרום לפסיקה ותעביר את המחשב למצב מיוחס (Privileged).
- b.** קריאת מערכת תגרום לפסיקה ותעביר את המחשב למצב משתמש (User).
- c.** קריאת מערכת תעביר את המחשב למצב מיוחס (Privileged) אך לא תגרום לפסיקה.
- d.** קריאת מערכת תעביר את המחשב למצב משתמש (User) אך לא תגרום לפסיקה.

מבחן מספר 2

1. בחרו את ההגדרה הנכונה ביותר עבור Interrupt Vector Table.

- a. טבלת קפיצות שמערכת ההפעלה הכינה בזמן האתחול. הטבלה מכילה כתובות שבעזרתם המעבד עובר לביצוע קוד שמטפל בפסיקות.
- b. טבלה שבהינתן אינדקס, נקבל את סוג הפסיקה.
- c. טבלה ששומרת את היסטוריית הפסיקות שהתרחשו מאז אתחול המחשב.
- d. טבלה ששומרת PID של תהליכים שעברו למצב שינה עקב פסיקה.

2. מה מכיל הקובץ "file.txt" בסיום התוכנית ומה התוכנית תדפיס?

```
int main(void) {
    char buf1[] = "1234ABCD5678\n";
    char buf2[] = "TEST";
    int fd, pos1, pos2, pos3, pos4;
    fd = open("file.txt", O_WRONLY | O_CREAT , S_IRUSR | S_IWUSR);
    write(fd, buf1, 12);
    pos1 = lseek(fd, -8, SEEK_END);
    fd = open("file.txt", O_WRONLY | O_TRUNC);
    write(fd, buf2, 4);
    pos2 = lseek(fd, 0, SEEK_CUR);
    pos3 = lseek(fd, -2, SEEK_CUR);
    write(fd, buf1, 3);
    pos4 = lseek(fd, 0, SEEK_CUR);
    printf("pos1= %d , pos2= %d , pos3= %d , pos4= %d \n", pos1, pos2, pos3, pos4);
}
```

- a. תדפיס 5 pos4= , pos3= 2 , pos2= 4 , pos1= 4 והקובץ מכיל: "TE123ST".
- b. תדפיס 4 pos4= , pos3= 1 , pos2= 4 , pos1= 3 והקובץ מכיל: "T123".
- c. תדפיס 5 pos4= , pos3= 2 , pos2= 4 , pos1= 4 והקובץ מכיל: "TE123".
- d. התוכנית קורסת.

3. סמנו את התשובה הנכונה עבור Superblocks.

- a. מכיל פרטים אודות כל מערכת הקבצים.
- b. מכיל קוד לאתחול המערכת.
- c. מערך של ביטים לניהול הבלוקים הפנויים.
- d. בלוק גדול לאחסון של מספר קבצים.

4. איך אפשר למנוע מהכותב לעבור למצב שינה כאשר ה-pipe מלא וקיים תהליך שסיים לקרוא?

- a. הקורא צריך לסגור את צד הקריאה והכותב צריך לסגור את צד הכתיבה.
- b. הקורא והכותב צריכים לסגור את צד הכתיבה.
- c. הקורא והכותב צריכים לסגור את צד הקריאה.
- d. כל התשובות נכונות.

5. מפת דפים הפוכה. נתון זיכרון פיסי של 1GB, גודל דף הוא 2kb, מה מספר הכניסות לטבלה?.

- a. 64kb
- b. 16kb
- c. 512kb
- d. 1MB

6. החלפת דף FIFO. נתונים 3 דפים פיסיים וסדרת מספרי הדפים:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

מה מספר הפסיקות לפי אלגוריתם FIFO (כולל פסיקות הכרחיות)?

- a. 12 פסיקות
- b. 15 פסיקות
- c. 11 פסיקות
- d. 9 פסיקות

7. מה החיסרון בניהול הבלוקים הפנויים באמצעות כמה רשימות? בחרו את התשובה הנכונה ביותר.

- a. שבירה פנימית ושבירה חיצונית.
- b. שבירה פנימית.
- c. שבירה חיצונית.
- d. הקצאה ושחרור בזמן לינארי.

8. האם התרחיש הבא אפשרי (בחרו את התשובה הנכונה ביותר):

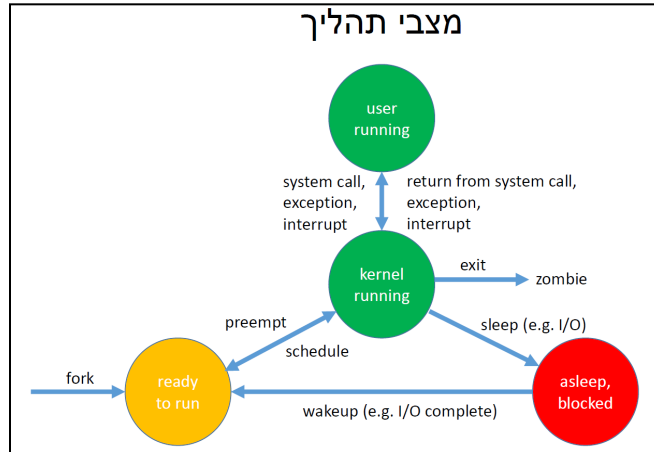
$TLB = miss, \quad Page\ Table = miss, \quad Cache = hit$

- a. אפשרי, לא כל דף חייב להיות ב-TLB אבל הוא בהכרח חייב להיות בטבלת הדפים. במידה והנתון היה לאחרונה בשימוש אז הוא גם נמצא במטמון.
- b. לא אפשרי, כל דף שנמצא ב-Page table הוא גם נמצא ב-TLB כי הוא מוכל בתוכו.
- c. לא אפשרי, הנתון לא יכול להתקיים במטמון אם הדף לא קיים בזיכרון.
- d. לא אפשרי, שאר התשובות המוצגות לא מדויקות.

9. מרחב הזיכרון (Address Space)

- a. קוד התכנית נטען לאזור הנקרא data, המשתנים שמוגדרים בתוך פונקציה יהיו באזור המחסנית.
- b. קוד התכנית נטען לאזור הנקרא text, המשתנים שמוגדרים בתוך פונקציה יהיו באזור המחסנית.
- c. קוד התכנית נטען לאזור הנקרא text, המשתנים שמוגדרים בתוך פונקציה יהיו באזור הנקרא data.
- d. קוד התכנית נטען לאזור המחסנית, המשתנים שמוגדרים בתוך פונקציה יהיו באזור הנקרא text.

10. מצבי תהליך (Process States) מי מצמד המשפטים שלהלן נכון ?



a. תהליך שמבקש קריאת נתון מקובץ, יעבור ממצב ריצה של משתמש (User Running) למצב המתנה למאורע (Blocked). תהליך שנמצא במצב המתנה למאורע (Blocked), לאחר שהמאורע יתרחש, יעבור למצב מוכן לריצה.

b. תהליך שמבקש קריאת נתון מקובץ, יעבור ממצב ריצה של משתמש (User Running) למצב ריצה של מערכת ההפעלה (Kernel Running). תהליך שנמצא במצב המתנה למאורע (Blocked), לאחר שהמאורע יתרחש, יעבור למצב ריצה.

c. תהליך שמבקש קריאת נתון מקובץ, יעבור ממצב ריצה של משתמש (User Running) למצב ריצה של מערכת ההפעלה (Kernel Running). תהליך שנמצא במצב המתנה למאורע (Blocked), לאחר שהמאורע יתרחש, יעבור למצב מוכן לריצה.

d. תהליך שמבקש קריאת נתון מקובץ, יעבור ממצב ריצה של משתמש (User Running) למצב מוכן לריצה. תהליך שנמצא במצב המתנה למאורע (Blocked), לאחר שהמאורע יתרחש, יעבור למצב ריצה.

מבחן מספר 3

1. סמנו את התשובה הנכונה. בפסיקות חיצוניות:

- a. ה-CPU יעשה בדיקה אם הגיע Interrupt לפני ביצוע הפקודה בשונה מפסיקה פנימית שבה ה-CPU יתפנה אחרי ביצוע הפקודה.
- b. ה-CPU יעשה בדיקה אם הגיע Interrupt רק אחרי ביצוע הפקודה בשונה מפסיקה פנימית שבה ה-CPU יעצור מיד ויתפנה למשימה החדשה הזאת.
- c. הפסיקות תלויות בריצת התוכנית בלבד.
- d. הפסיקות נוצרו על ידי Exceptions כמו למשל זיכרון לא נגיש או חישוב חלוקה ב-0.

2. נתון קובץ טקסט file.txt ובתוכו כתוב: "Hi there!". נתון הקוד הבא. בחרו את התשובה הנכונה.

```
int main() {  
    int file_desc = open("file.txt", O_WRONLY | O_APPEND);  
    dup2(file_desc, 1) ;  
    printf("How are you?\n");  
    return 0;  
}
```

- a. התוכנית תדפיס Hi there!How are you? ובקובץ כתוב Hi there!
- b. התוכנית לא תדפיס כלום ובקובץ כתוב Hi there!
- c. התוכנית תדפיס How are you? ובקובץ כתוב Hi there!\nHow are you?
- d. התוכנית לא תדפיס כלום ובקובץ כתוב Hi there!\nHow are you?

3. נתון קוד עבור הרשאות לקובץ myfile. סמנו את התשובה הנכונה.

```
chmod 754 myfile
```

מה ההרשאות עבור group ?

- a. read
- b. write
- c. read, execute
- d. write, execute

4. מפת דפים הפוכה. נתון זיכרון פיסי של 4GB, גודל דף הוא 4kb, מה מספר הכניסות לטבלה?.

- a. 64kb
- b. 16kb
- c. 512kb
- d. 1MB

5. נתון קובץ הטקסט הבא (fruits.txt):

```
banana  
apple  
mango  
watermelon  
pineapple
```

משתמש הכניס את פקודת ה-shell הבאה: `cat fruits.txt | sort -r | head -1`
מה תדפיס הפקודה?

a. watermelon

b. pineapple

c. banana

d. תחזיר את כל הפירות ממוינים בסדר לקסיקוגרפי.

6. החלפת דף LRU. נתונים 3 דפים פיסיים וסדרת מספרי הדפים:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

מה מספר הפסיקות לפי אלגוריתם LRU (כולל פסיקות הכרחיות)?

a. 12 פסיקות

b. 15 פסיקות

c. 11 פסיקות

d. 9 פסיקות

7. דליפת זיכרון. השלימו את השורה החסרה (שורה 3).

```
1 void leak1() {  
2     Object *x = new Object;  
3  
4     return;  
5 }
```

a. `free(x);`

b. `x = new Object;`

c. `free(*x);`

d. `x = NULL;`

8. מטמון במיפוי ישיר. נתון שכל בלוק בגודל 8 בתים. במטמון יש 8 בלוקים. נתונה הכתובת הדצימלית 25.

מה מספר הבלוק במטמון ומה מספר הבלוק בזיכרון הראשי?

a. `Main = 3, Cache = 3`

b. `Main = 2, Cache = 8`

c. `Main = 8, Cache = 2`

d. `Main = 25, Cache = 8`

9. החלפת תהליך (Context Switch)

- a.** בזמן החלפת תהליך, צריך לשמור את האוגרים של התהליך המוחלף, בקובץ.
- b.** בזמן החלפת תהליך, צריך לשמור את הקבצים שהתהליך המוחלף פתח, בדיסק.
- c.** בזמן החלפת תהליך, צריך לשמור את הקבצים שהתהליך המוחלף פתח, ב-PCB.
- d.** בזמן החלפת תהליך, צריך לשמור את האוגרים של התהליך המוחלף, ב-PCB.

10. יצירת תהליך (fork())

- a.** כשיוצרים תהליך באמצעות fork() צריך להעתיק את הקבצים שההורה פתח, מההורה לילד.
- b.** כשיוצרים תהליך באמצעות fork() ה-PCB של הילד שווה ל-PCB של ההורה.
- c.** כשיוצרים תהליך באמצעות fork() קוד התכנית של הילד שווה לקוד התכנית של ההורה.
- d.** כשיוצרים תהליך באמצעות fork() הערך המוחזר מ fork() שווה אצל ההורה ואצל הילד.

מבחן מספר 4

1. מהו הערך שחוזר מפונקציית הקריאה `open()`?
- a. הפונקציה מחזירה את המספר 1 אם פתיחת הקובץ הצליחה או 0 אם קרתה שגיאה.
 - b. הפונקציה לא מחזירה כלום.
 - c. הפונקציה מחזירה את המסלול אל הקובץ, במידה וקרתה שגיאה תחזיר NULL.
 - d. הפונקציה מחזירה את האינדקס של הקובץ שנפתח מתוך ה-File Descriptor Table.
2. לדיסק יש את הנתונים הבאים:
- גודל הסקטור - Sector size: 512 bytes
 - מספר המשטחים - Surfaces: 4
 - זמן החיפוש הממוצע - T_seek : 5 ms
 - מספר הסיבובים לדקה - Rotational rate: 10,000 RPM
 - מספר הסקטורים בכל מסילה - Number of sectors/track: 1,000
- כמה זמן ייקח לקרוא קובץ בגודל 5MB כלומר מתחילתו ועד סופו, כאשר הבלוקים מסודרים בצורה הטובה ביותר.
- a. 69.44ms
 - b. 12.7ms
 - c. 71.2ms
 - d. 8.006ms
3. הרשאת `set-user-id`. סמנו את התשובה הנכונה.
- a. המשתמש יכול לשנות את ההרשאות של הקובץ עם `syscall` שנקרא `chmod()`.
 - b. משתמש יכול להריץ קובץ תכנית ששייך למשתמש אחר, אם המשתמש האחר נתן לו הרשאה.
 - c. כדי לפתוח, ליצור ולמחוק קובץ יש צורך בהרשאת ביצוע, כלומר מעבר בכל התיקיות מהשורש ועד לקובץ הנתון.
 - d. זו הרשאה מיוחדת שמתייחסת רק עבור תיקיות.
4. שימוש ב-`pipe` לחיבור מסננים (`sort`, `uniq` וכו'...). ל-`pipe` מחוברים תהליך האב (כותב) עם `fd=4` ותהליך הילד (קורא) עם `fd=3`. איך הקריאה מ-0 (`stdin`) והכתיבה ל-1 (`stdout`) תקרא ותכתוב ל-`pipe` (שהם 3 ו-4)? סמנו את התשובה הנכונה ביותר.
- a. נעתיק באמצעות `dup2` את ה-`fd` (מצביעים) של ה-`pipe` ל-`fd` של `stdout` ושל `stdin`.
 - b. נשכפל באמצעות `fork` נוסף את ה-`fd` (מצביעים) של ה-`pipe` ל-`fd` של `stdout` ושל `stdin`.
 - c. נשתמש ב-`write` ו-`open` כדי לכתוב לקובץ ואז נשלח לתהליך הקורא.
 - d. כל התשובות לא נכונות.

5. טבלת גיבוב למפת דפים הפוכה.

- a.** כל דף לוגי ממופה לדף פיזי באמצעות פונקציית גיבוב פשוטה.
- b.** ה-MMU יחפש תחילה במטמון, כאשר המיפוי לא נמצא במטמון יתבצע חיפוש בטבלת הגיבוב ויוכנס למטמון.
- c.** יתכן שפונקציית הגיבוב תמפה כמה דפים לוגיים לאותו דף פיזי, הדפים הנוספים משורשרים לדפים הפנויים הבאים.
- d.** כל התשובות נכונות.

6. אלגוריתם השעון. נניח שישנם 4 דפים פיזיים המכילים את ארבעת הדפים הלוגיים 1,2,3,4 לפי הסדר.

המחוג מצביע על דף מספר 1. התכנית המשיכה לגשת לסדרת הדפים (משמאל לימין): 5,3,4,1,6. מה מספר הפסיקות לפי אלגוריתם השעון (החל מהמשך התוכנית - לא כולל הכרחיות).

- a.** 1
- b.** 2
- c.** 3
- d.** 4

7. מה עלול לקרות בשימוש מצביע dangling?

- a.** קריאה מהמצביע עלולה להחזיר תוכן אקראי.
- b.** כתיבה למצביע עלולה לשנות תוכן של אובייקט אחר.
- c.** שחרור המצביע עלולה לשחרר אובייקט אחר או לשבש את הערימה.
- d.** כל התשובות נכונות.

8. נניח שגודל הבלוק הוא 16 בתים והמטמון מכיל 64 בלוקים, לאיזה בלוק במטמון ימופה בית שכתובתו 1210?

- a.** $Main = 2, Cache = 2$
- b.** $Main = 75, Cache = 11$
- c.** $Main = 11, Cache = 75$
- d.** $Main = 64, Cache = 11$

9. הפנית קלט/פלט (redirection). הפקודה `ls | wc`

- a. כותבת את הפלט של הפקודה `ls` לקובץ `wc`.
- b. מנתבת את הפלט של הפקודה `ls` לקלט של הפקודה `wc`.
- c. מנתבת את הפלט של הפקודה `wc` לקלט של הפקודה `ls`.
- d. כותבת את הפלט של הפקודה `ls` לסוף הקובץ `wc`.

10. יצירת תהליך (`fork()`). בתיקיה `"home/user/mydocs/"` ישנם שלשה קבצים `a`, `b` ו-`c`. מהו הפלט של התכנית הבאה?

```
int value = 5;
int main()
{
    pid_t pid;
    pid = fork();
    if (pid == 0) {
        value += 15;
        execlp("/bin/ls", "ls", "/home/user/mydocs", NULL);
        printf("%d\n", value);
        return 0;
    }
    else if (pid > 0) {
        wait(NULL);
        printf("%d\n", value);
        return 0;
    }
}
```

a. a, b, c
20

b. a, b, c
5

c. a, b, c
5
20

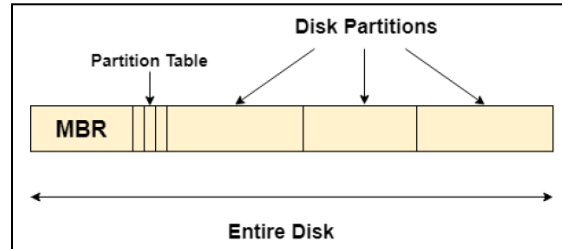
d. 5
a, b, c

מבחן מספר 5

1. כאשר תהליך חדש נוצר, מערכת ההפעלה פותחת עבורו בטבלת ה-File Descriptor Table את הקבצים:

- a. STDIN, STDOUT, STDERR
- b. STDIN, STDOUT
- c. STDIN, STDERR
- d. המערכת לא פותחת עבורו כלום כי התהליך עוד לא התחיל לעבוד.

2. חלוקת הדיסק למחיצות. נתונה תמונה שהוצגה בהרצאה. סמנו את התשובה הנכונה.



- a. סקטור 0 של דיסק נקרא MBR ומכיל קוד שמאתחל את המחשב.
- b. Partition Table היא טבלה המפרטת את כתובות ההתחלה והסיום של כל מחיצה ודגל המציין את המחיצה הפעילה (מאיזו מחיצה יש לאתחל את המערכת).
- c. בכל מחיצה אפשר ליצור מערכת קבצים נפרדת.
- d. כל התשובות נכונות.

3. הרשאת sticky bit. סמנו את התשובה הנכונה.

- a. אם מוסיפים לתיקיה הרשאת sticky bit אז משתמש רגיל לא יכול לקרוא את התיקיה.
- b. אם מוסיפים לתיקיה הרשאת sticky bit אז משתמשים בפקודות קריאה וכתובה של קבצים בתיקיה.
- c. אם מוסיפים לתיקיה הרשאת sticky bit אז המשתמש יכול להריץ קובץ תכנית ששייך למשתמש אחר.
- d. אם מוסיפים לתיקיה הרשאת sticky bit אז המשתמש יוכל למחוק רק קבצים השייכים לו.

4. חלוקה לדפים. בחרו את התשובה הנכונה.

- a. הזיכרון הפיסי (RAM) מחולק לחלקים בגודל שווה המכונים דפים.
- b. הזיכרון הלוגי של כל תהליך מחולקים לדפים כאשר הגודל של כל דף שונה או שווה לשאר.
- c. כשהמעבד מוציא כתובת לוגית לזיכרון, החומרה (MMU) משתמשת בטבלת הדפים של התהליך כדי לתרגם את הכתובת הלוגית לפיזית.
- d. הזיכרון הלוגי של כל תהליך מחולקים למסגרות כאשר הגודל של כל דף שונה או שווה לשאר.

5. זיכרון וירטואלי. במחשב 32bit אפשר להגיע ל- $4GB = 2^{32}$ כתובות. כמה זיכרון וירטואלי יקבל כל תהליך?.

- a. 1GB
- b. 32GB
- c. 4GB
- d. 256MB

6. החלפת דף - אלגוריתם השעון. נתונים 3 דפים פיסיים וסדרת מספרי הדפים:

2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2

מה מספר הפסיקות לפי אלגוריתם השעון (כולל פסיקות הכרחיות)?

- a. 12 פסיקות
- b. 8 פסיקות
- c. 11 פסיקות
- d. 9 פסיקות

7. בעיות בניהול ידני של זיכרון דינמי. איזה סוג של בעיה קיים בקוד הבא?

```
1 for (p = head; p != NULL; p = p-> next) {  
2     free (p) ;  
3 }
```

- a. דליפת זיכרון (leak).
- b. מצביע משוחרר (dangling).
- c. בעיית reference cycles.
- d. אובדן sticky bits.

8. נתונה הכתובת 0x00003004. ונתונים $b = 2$, $s = 3$, $t = 27$. (offset, index, tag) חשבו את x =המקום של הבית בתוך הבלוק. y =מספר הבלוק במטמון. z =מספר הבלוק בזיכרון.

- a. $x = 0$, $y = 1$, $z = 3004$
- b. $x = 0$, $y = 1$, $z = 384$
- c. $x = 4$, $y = 3$, $z = 3004$
- d. $x = 2$, $y = 3$, $z = 27$

9. תהליכים וחוטמים (Processes and Threads)

- a. קוד התכנית משותף לכל החוטמים (threads), אך לכל חוט (thread) יש מחסנית משלו וקבצים פתוחים משלו.
- b. המחסנית והקבצים הפתוחים משותפים לכל החוטמים (threads), אך לכל חוט (thread) יש קוד תכנית משלו.
- c. הקבצים הפתוחים משותפים לכל החוטמים (threads), אך לכל חוט (thread) יש קוד תכנית ומחסנית משלו.
- d. קוד התכנית והקבצים הפתוחים משותפים לכל החוטמים (threads), אך לכל חוט (thread) יש מחסנית משלו.

10. שיטות להקצאת קבצים (File Allocation Methods)

- a.** בהקצאה רציפה (Contiguous File Allocation) יש יתרון כי קריאה של קובץ היא יותר מהירה, ויש חיסרון כי קשה לנצל את כל הבלוקים בדיסק.
- b.** בהקצאה רציפה (Contiguous File Allocation) יש יתרון כי יש ניצול של כל הבלוקים בדיסק, ויש חיסרון כי קריאה של קובץ היא פחות מהירה.
- c.** בהקצאה רציפה (Contiguous File Allocation) יש יתרונות כי יש ניצול של כל הבלוקים בדיסק וקריאה של קובץ היא יותר מהירה.
- d.** בהקצאה רציפה (Contiguous File Allocation) יש חסרונות כי קריאה של קובץ היא פחות מהירה וקשה לנצל את כל הבלוקים בדיסק.

מבחן מספר 6

- 1.** בחרו את התשובה הנכונה עבור Kernel buffering of file I/O.
- a.** בקריאה וכתיבה לדיסק, הפונקציות read, write לא קוראות או כותבות ישירות לדיסק אלא הן קוראות מה-buffer של הקרנל כדי לחסוך זמן קריאה או כתיבה לדיסק.
 - b.** כל buffer בקרנל הוא בגודל בלוק של דיסק.
 - c.** מערכת ההפעלה כותבת מידי פעם את ה-buffers לדיסק באמצעות הפקודה `sync()` כדי לא לאבד מידע במקרה של קריסת מערכת.
 - d.** כל התשובות נכונות.
- 2.** FTL - Flash Translation Layer. סמנו את התשובה הנכונה.
- a.** בקר ה-FTL מקבל פקודות ממערכת ההפעלה לקריאה וכתיבה של דפים פיסיים ומתרגם את הפקודות לקריאה, מחיקה וכתיבה לדיסק של דפים לוגיים.
 - b.** בקר ה-FTL מקבל פקודות ממערכת ההפעלה לקריאה וכתיבה של דפים לוגיים ומתרגם את הפקודות לקריאה, מחיקה וכתיבה לדיסק של דפים פיסיים.
 - c.** תפקידו לבקר שחיקה של בלוקים בזיכרון מסוג flash.
 - d.** כל התשובות לא נכונות.
- 3.** קריאת המערכת `stat()` סמנו את התשובה הנכונה.
- a.** פונקציה, המופעלת ע"י `syscall`, מחזירה לנו מבנה שנקרא `stat` שמכיל מידע לגבי הקובץ.
 - b.** פונקציה, המופעלת ע"י `syscall`, מחזירה לנו מבנה שנקרא `stat` שמכיל היסטוריה על שינויים בתיקייה.
 - c.** פונקציה, המופעלת ע"י `syscall`, מחזירה לנו log בפורמט `(txt)` שמכיל סטטיסטיקות ביצוע של תהליך.
 - d.** פונקציה, המופעלת ע"י `syscall`, מחזירה לנו את הזמן הממוצע של בקשות התהליך לקרנל `(float)`.
- 4.** נתונים כתובת בגודל 16bit וגודל דף הוא $1k = 1024$ בתים. מה גודל התוכנית המקסימלית?
- a.** 64kb
 - b.** 16kb
 - c.** 32kb
 - d.** 86kb

5. סמנו את ההגדרה הנכונה עבור שיטת ה-Demand Paging.

- a. מערכת ההפעלה תספק לתהליך רק את הדפים הנחוצים אליו במקום לטעון את כל הדפים מראש.
- b. מערכת ההפעלה תספק לתהליך את כל הדפים מראש.
- c. שיטה להעתקת דפים של תהליך בעת שימוש ב-fork.
- d. שיטה להעתקת דפים של תהליך בעת שימוש ב-dup.

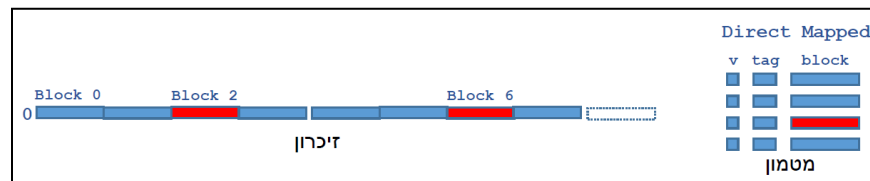
6. שיטת COW - copy-on-write. מה הבעיה ש-COW פותרת?

- a. הילד מבצע exec מיד אחרי ה-fork ומחליף את קטעי הזיכרון שהעתיק מאביו ולכן ביצוע ההעתקה היה מיותר.
- b. משתמש לא יכול לכתוב לקובץ כי אין לו הרשאה לכך ואז נוצר deadlock.
- c. כשאין מקום בזיכרון הפיזי אז נדרסים נתונים שיכולים להיות חשובים לתוכנית.
- d. כל התשובות לא נכונות.

7. יחס הפגיעה במטמון. תכנית ביצעה 3500 פקודות הניגשות לזיכרון לקריאה או כתיבה של נתונים. 3098 מהגישות נמצאו במטמון. מהו יחס הפגיעה וההחטאה של הגישות לנתונים?

- a. $HitRate = 0.8, MissRate = 0.2$
- b. $HitRate = 0.885, MissRate = 0.115$
- c. $HitRate = 0.625, MissRate = 0.375$
- d. $HitRate = 0.99, MissRate = 0.001$

8. נתבונן בתמונה הבאה. נניח שבבוק 6 ו-2 בזיכרון (באדום) ממופים לאותו הבלוק במטמון (באדום).



איזו בעיה קיימת בתרחיש הזה?

- a. בעיית Cold miss
- b. בעיית Conflict miss
- c. בעיית Capacity miss
- d. לא קיימת כאן בעיה.

9. פתיחת קובץ (File open). כדי לפתוח קובץ לכתיבה בלבד, אם הקובץ קיים לאפס אותו, אם הקובץ לא קיים ליצור אותו, נשתמש בדגלים (flags) הבאים:

- a. $WRONLY | O_TRUNC | O_APPEND$
- b. $O_CREAT | O_RDWR | O_TRUNC$
- c. $O_CREAT | O_WRONLY | O_TRUNC$
- d. $O_CREAT | O_WRONLY | O_APPEND$

10. lseek. פתחנו קובץ בגודל 1000 בתים וקראנו ממנו 100 בתים, לאחר מכן בצענו את הפקודות:

```
lseek(fd, 10, SEEK_SET);  
pos = lseek(fd, 0, SEEK_CUR);
```

מה הערך שנכנס ל-pos?

a. 109

b. 110

c. 10

d. 1010

מבחן מספר 7

1. נתונות 2 קטעי קוד. סמנו את התשובה הנכונה ביותר.

```
void copy1() {
    char c;
    int in, out;
    in = open("file.in", O_RDONLY);
    out = open("file.out", O_WRONLY|O_CREAT, S_IRUSR|S_IWUSR);
    while(read(in,&c,1) == 1)
        write(out,&c,1);
}

void copy2() {
    char block[2048];
    int in, out;
    int nread;
    in = open("file.in", O_RDONLY);
    out = open("file.out", O_WRONLY|O_CREAT, S_IRUSR|S_IWUSR);
    while((nread = read(in,block,sizeof(block))) > 0)
        write(out,block,nread);
}
```

- a. הפונקציה copy1 יותר מהירה מהפונקציה copy2 כיוון שאנחנו קוראים רק תו אחד בלבד.
- b. הפונקציה copy1 יותר איטית מהפונקציה copy2 כיוון שאפשר להעתיק בכל שלב רק גודל 1.
- c. הפונקציה copy2 תקרוס כיוון שהפונקציה read() לא יודעת לקבל מערך של תווים.
- d. הפונקציה copy1 לא נכונה כיוון ש read() לא תחזיר fd=1 כי הוא תפוס מראש (STDOUT).

2. בהרצאה למדנו על RAID-4 (זוגיות). בכל כתיבה של בלוק נתונים צריך לעדכן את בלוק הזוגיות.

נתונים: $C_{old} = 0010$, $C_{new} = 1001$, $P_{old} = 1110$.

חשבו את בלוק הזוגיות P_{new} לאחר עדכון של כתיבת בלוק וסמנו את התשובה הנכונה.

a. $P_{new} = 0101$

b. $P_{new} = 1101$

c. $P_{new} = 1110$

d. $P_{new} = 0111$

3. מערכות קבצים Journaling. סמנו את התשובה הנכונה.

- a. כדי למנוע מצב שבו מערכת הקבצים מעודכנת באופן חלקי, כותבים את כל העדכונים לדיסק ורק אחר כך מבצעים כתיבה ל-log.
- b. כדי למנוע מצב שבו מערכת הקבצים מעודכנת באופן חלקי, כותבים את כל העדכונים ל-log ורק אחר כך מבצעים כתיבה לדיסק.
- c. כדי שמערכת הקבצים לא תשתבש, כתיבה וקריאה של תיקיה מתבצעות באמצעות הקרנל.
- d. כל התשובות לא נכונות.

4. תרגום כתובת לוגית לפיסית בחלוקה לדפים. נתונים כתובות בגודל 32bit וגודל דף הוא $4k = 4096$ בתים. כמה ביטים אנחנו צריכים כדי לציין את מספר הדף?

- a. 32bit
- b. 16bit
- c. 8bit
- d. 20bit

5. באיזה חלק בזיכרון נשמר קטע הקוד המסומן באדום?

```
#include <stdio.h>
#include <stdlib.h>
int x;
int y = 15;
int main(int argc, char *argv[]) {
    int *values;
    int i;
    values = (int *)malloc(sizeof(int)*5);
    for(i = 0; i < 5; i++)
        values[i] = i;
    return 0;
}
```

- a. במחסנית (stack)
- b. בערימה (heap)
- c. text-ב
- d. initialized data-ב

6. תכנית תקצה אובייקטים בערימה (heap):

- a. אם גודל האובייקט ידוע מראש.
- b. כדי למנוע שיתוף של נתונים עבור כמה פונקציות באמצעות מצביעים.
- c. כדי לאפשר שיתוף של נתונים עבור כמה פונקציות באמצעות מצביעים.
- d. אם לא נשאר מקום במחסנית.

7. איזה בעיה עלולה לקרות בשימוש אלגוריתם Mark and Sweep בשפות ++C?

- a. נמחוק ספריות של מערכת ההפעלה.
- b. נמחוק נתונים שאינם garbage.
- c. כל נתון שיפונה הוא garbage אבל לא כל ה-garbage יפונה.
- d. כל התשובות נכונות.

8. נתון קטע קוד. נניח שמטמון במיפוי ישיר מכיל שני בלוקים כל אחד בגודל 16 בתים.

```
1 float dotprod(float x[8], float y[8]) {  
2     float sum = 0.0;  
3     int i;  
4     for(i = 0; i < 8; i++) {  
5         sum += x[i]*y[i];  
6     }  
7     return sum;  
8 }
```

המערך x תופס $32 = 8 \cdot 4$ בתים ומתחיל בכתובת 0.
המערך y תופס $32 = 8 \cdot 4$ בתים ומתחיל מיד אחרי x.
מה יהיה אחוז ההחטאה (miss)?

- a. 25%
- b. 50%
- c. 75%
- d. 100%

9. chmod

לקובץ myfile יש את ההרשאות הבאות:

rw-rw-rw-

איזו פקודת chmod תשנה את ההרשאות ל:

rw-rw-r--

- a. chmod u+x,g-w myfile
- b. chmod u+x,o-w myfile
- c. chmod u=x,o=w myfile
- d. chmod u+rw,x,o+r myfile

10. symbolic link. אם הקובץ b הוא symbolic link לקובץ a:

- a. b יכיל את השביל (file path) לקובץ a.
- b. לקבצים a ו-b יהיה אותו מספר inode.
- c. הקבצים a ו-b תמיד יהיו באותה תיקיה.
- d. לקבצים a ו-b תמיד יהיו באותה מערכת קבצים.

מבחן מספר 8

1. סמנו את התשובה הנכונה. (לברר).
- a. מטבלת File Descriptor לטבלת Open File המצביעים בצורה של "Many to one" בעוד שמטבלת Open File לטבלת ה-I-Node המצביעים בצורה של "One to many".
 - b. שיטת שלושת הטבלאות File Descriptor, Open File, I-Node לא יעילה.
 - c. מטבלת File Descriptor לטבלת Open File המצביעים בצורה של "One to many" בעוד שמטבלת Open File לטבלת ה-I-Node המצביעים בצורה של "Many to one".
 - d. כל התשובות לא נכונות.
2. הקצאה עם מצביעים. נניח שגודל בלוק הוא 4k ומצביע לבלוק הוא בגודל 32 ביטים. אם יש 12 מצביעים ישירים ונניח שיש רק מצביע אחד (לא ישיר) לבלוק של מצביעים, מה גודל הקובץ המקסימלי?
- a. גודל הקובץ המקסימלי הוא 4,000k
 - b. גודל הקובץ המקסימלי הוא 4,144k
 - c. גודל הקובץ המקסימלי הוא 3,982k
 - d. גודל הקובץ המקסימלי הוא 41,400k
3. מה היתרון בשימוש Metadata journaling? סמנו את התשובה הנכונה.
- a. גרסה קלה יותר. השיטה הזאת משתמשת בדחיסת נתונים כדי להקטין את גודל הלוג.
 - b. כדי לחסוך בהעתקה כפולה לדיסק, אפשר להעתיק את הבלוקים של ה-metadata (כלומר רק את ה-inode, bitmap) ישירות לדיסק ולאחר מכן לכתוב ל-log רק את ה-data.
 - c. כדי לחסוך בהעתקה כפולה לדיסק, אפשר להעתיק את הבלוקים של ה-data ישירות לדיסק ולאחר מכן לכתוב ל-log רק את ה-metadata (כלומר רק את ה-inode, bitmap).
 - d. אין יתרון בשימוש בשיטה הזאת.
4. יחס הפגיעה במטמון (TLB hit ratio). נניח שזמן הגישה לזיכרון הוא 10 ננו-שניות ויחס הפגיעה או 92%. מה יהיה זמן הגישה האפקטיבי (בננו-שניות)?
- a. 10.8 ננו-שניות.
 - b. 10.1 ננו-שניות.
 - c. 10 ננו-שניות.
 - d. 82 ננו-שניות.

5. באיזה חלק בזיכרון נשמר קטע הקוד המסומן באדום?

```
#include <stdio.h>
#include <stdlib.h>
int x;
int y = 15;
int main(int argc, char *argv[]) {
    int *values;
    int i;
    values = (int *)malloc(sizeof(int)*5);
    for(i = 0; i < 5; i++)
        values[i] = i;
    return 0;
}
```

- a. במחסנית (stack)
- b. בערימה (heap)
- c. ב-text
- d. ב-initialized data

6. מה מחזירה הפונקציה malloc()?

- a. 1- עבור שגיאה או 0 אם ההקצאה בוצעה בהצלחה.
- b. מצביע סתמי (void) לתחילת בלוק זיכרון בגודל שביקשנו.
- c. גודל הזיכרון הכולל לאחר ההקצאה.
- d. לא תחזיר כלום.

7. נתונים: $t_{Memory} = 100 \text{ cycles}$, $t_{cache} = 2 \text{ cycles}$, $MissRate = 0.375$.

מה זמן הגישה הממוצע לנתון בזיכרון?

- a. 39.5 cycles
- b. 39 cycles
- c. 38.5 cycles
- d. 24.3 cycles

8. נתון קטע קוד. נניח שמטמון במיפוי ישיר מכיל 3 בלוקים כל אחד בגודל 16 בתים.

```
1 float dotprod(float x[12], float y[12]) {
2     float sum = 0.0;
3     int i;
4     for(i = 0; i < 12; i++) {
5         sum += x[i]*y[i];
6     }
7     return sum;
8 }
```

המערך x תופס $12 \cdot 4 = 48$ בתים ומתחיל בכתובת 0.

המערך y תופס $12 \cdot 4 = 48$ בתים ומתחיל מיד אחרי x.

מה יהיה אחוז ההחטאה (miss)?

- a. 25%
- b. 50%
- c. 75%
- d. 100%

9. הרשאות לקבצים (File Permissions)

התיקייה C נמצאת בתוך התיקייה B שנמצאת בתוך התיקייה A שנמצאת בתוך תיקיית השורש, כך:

A/B/C/ מהן ההרשאות הדרושות כדי למחוק קובץ מהתיקייה C?

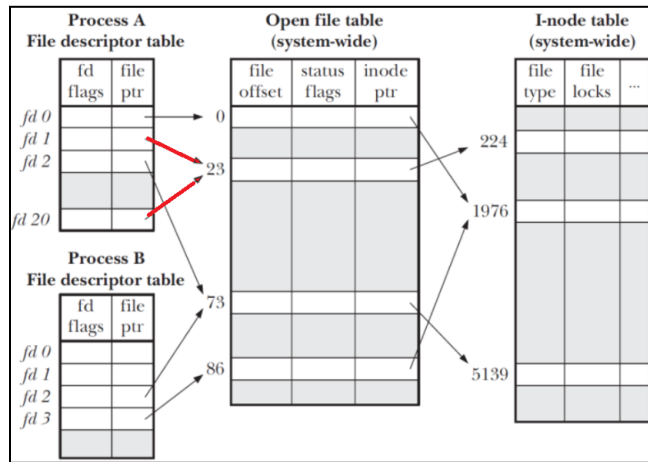
- a. הרשאת ביצוע לתיקיות C, ו-A, B, וכן הרשאת כתיבה ל-C.
- b. הרשאת קריאה לתיקיית השורש ולתיקיות C, ו-A, B, וכן הרשאת כתיבה ל-C.
- c. הרשאת ביצוע לתיקיית השורש ולתיקיות C, ו-A, B, וכן הרשאת כתיבה ל-C.
- d. הרשאת קריאה לתיקיות C, ו-A, B, וכן הרשאת כתיבה ל-C.

10. זמנים של קובץ. מהם שלושת הזמנים שנשמרים עבור כל קובץ במערכת unix?

- a. זמן היצירה של הקובץ, זמן הכתיבה האחרון של הקובץ, זמן הקריאה האחרון של הקובץ.
- b. זמן הכתיבה האחרון של הקובץ, זמן הקריאה האחרון של הקובץ, זמן הקריאה האחרון של תכונות הקובץ (כגון הרשאות).
- c. זמן היצירה של הקובץ, זמן הכתיבה האחרון של הקובץ, זמן השינוי האחרון של תכונות הקובץ (כגון הרשאות).
- d. זמן הכתיבה האחרון של הקובץ, זמן הקריאה האחרון של הקובץ, זמן השינוי האחרון של תכונות הקובץ (כגון הרשאות).

מבחן מספר 9

1. נתונה תמונה שהוצגה בהרצאה. איזה פעולה מאפשרת את התרחיש שמתואר במצביעים האדומים?



- a. שימוש בפונקציית `fork()`.
- b. שימוש בפונקציית `dup()`.
- c. שימוש בפונקציית `open()`.
- d. כל התשובות נכונות.

2. נניח שגודל בלוק הוא 8k ומצביע לבלוק הוא בגודל 4 בתים. inode מכיל 12 מצביעים ישירים (לבלוקים של data) ונניח שיש רק מצביע אחד (לא ישיר) לבלוק של מצביעים, מה גודל הקובץ המקסימלי (בבתים) של קובץ במערכת קבצים זו?

- a. גודל הקובץ המקסימלי הוא 16,000k.
- b. גודל הקובץ המקסימלי הוא 4,144k.
- c. גודל הקובץ המקסימלי הוא 16,480k.
- d. גודל הקובץ המקסימלי הוא 8,192k.

3. קריאת המערכת `exec()` סמנו את התשובה הנכונה.

- a. מערכת ההפעלה תשחרר את קטעי הזיכרון של התוכנית הקודמת.
- b. תהליך הילד יבצע `exec()` כדי להחליף את התוכנית שהוא מריץ בתוכנית אחרת.
- c. הקריאה ל-`exec()` לא חוזרת (הקוד שקרא ל-`exec()` לא קיים).
- d. כל התשובות נכונות.

4. טבלת הדפים הגדולה. נניח מחשב 32bit וגודל דף 8k. כל כניסה בטבלת הדפים היא בגודל 4 בתים. כמה זיכרון תתפוס טבלת הדפים בזיכרון הפיסי עבור כל תהליך?.

- a. 4MB
- b. 1MB
- c. 2MB
- d. 512k

5. באיזה חלק בזיכרון נשמר קטע הקוד המסומן באדום?

```
#include <stdio.h>
#include <stdlib.h>
int x;
int y = 15;
int main(int argc, char *argv[]) {
    int *values;
    int i;
    values = (int *)malloc(sizeof(int)*5);
    for(i = 0; i < 5; i++)
        values[i] = i;
    return 0;
}
```

- a. במחסנית (stack)
- b. בערימה (heap)
- c. ב-text
- d. ב-initialized data

6. הקצאת בלוקים פנויים. כאשר תכנית מבקשת הקצאה בגודל מסוים, הזיכרון המוקצה צריך להיות בגודל שווה או גדול מהגודל המבוקש. מה החיסרון של שיטת Next fit?

- a. זמן חיפוש הכי ארוך מבין כל השיטות.
- b. יוצר בלוקים קטנים בתחילת הרשימה.
- c. מותיר בלוקים גדולים בסוף הרשימה.
- d. לא מותיר בלוקים גדולים בסוף הרשימה.

7. נתונים: $t_{L1} = 1 \text{ cycle}$, $t_{L2} = 10 \text{ cycles}$, $t_{Memory} = 100 \text{ cycles}$, $Miss_{L1} = 5\%$, $Miss_{L2} = 20\%$. מה זמן הגישה הממוצע לזיכרון כאשר יש שתי רמות של מטמון?

- a. 39.5 cycles
- b. 9 cycles
- c. 2.5 cycles
- d. 2.3 cycles

8. נתון קטע קוד. נניח שמטמון במיפוי ישיר מכיל 32 בלוקים כל אחד בגודל 16 בתים.

```
1 int func() {  
2     int x[2][128];  
3     int i, sum = 0;  
4     for (i = 0; i < 128, i++) {  
5         sum += x[0][i] * x[1][i];  
6     }  
7     return sum;  
8 }
```

המערך x תופס $1024 = 4 \cdot 128 \cdot 2$ בתים ומתחיל בכתובת 0.

מה יהיה אחוז הפגיעה (hit)?

- a. 0%
- b. 25%
- c. 75%
- d. 100%

9. מערכת הקבצים של unix. במערכת הקבצים הרגילה של unix, הקצאת הבלוקים היא בשיטת:

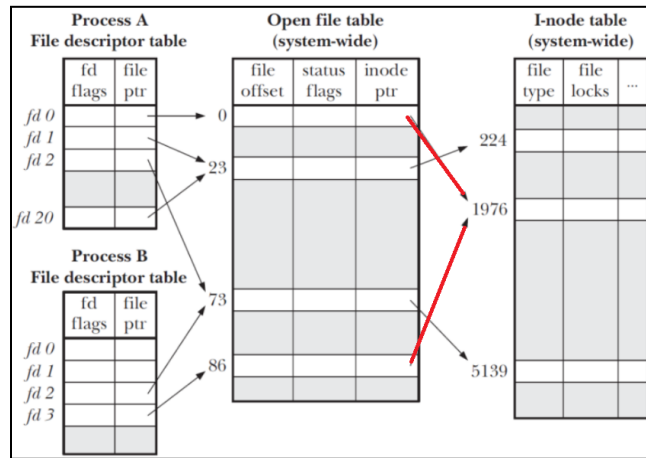
- a. הקצאה משורשרת של בלוקים (Linked-List Allocation)
- b. הקצאה משורשרת בשיטת FAT (File Allocation Table)
- c. בלוק שמצביע לבלוקים של הקובץ (Indexed Allocation)
- d. הקצאה רציפה של בלוקים (Contiguous File Allocation)

10. RAID. כאשר מחברים דיסקים בשיטת RAID level 0:

- a. קריאת קובץ תהיה יותר מהירה, אם דיסק אחד קורס המידע לא הולך לאיבוד.
- b. קריאת קובץ תהיה יותר איטית, אך אם דיסק אחד קורס המידע לא הולך לאיבוד.
- c. קריאת קובץ תהיה יותר מהירה, אך אם דיסק אחד קורס המידע הולך לאיבוד.
- d. המידע נכתב על שני דיסקים (mirror), אם דיסק אחד קורס המידע לא הולך לאיבוד.

מבחן מספר 10

1. נתונה תמונה שהוצגה בהרצאה. איזה פעולה מאפשרת את התרחיש שמתואר במצביעים האדומים?



- a. שימוש בפונקציית `fork()`.
- b. שימוש בפונקציית `dup()`.
- c. שתי קבצים מצביעים על אותה תיקייה (נמצאים באותה התיקייה).
- d. כל התשובות נכונות.

2. נניח שגודל בלוק הוא 4k ומצביע לבלוק הוא בגודל 4 בתים. inode מכיל 12 מצביעים ישירים (לבלוקים של data) ונניח שיש רק מצביע אחד (triple indirect blocks) לבלוק של מצביעים, מה גודל הקובץ המקסימלי (בבתים) של קובץ במערכת קבצים זו?

- a. גודל הקובץ המקסימלי הוא 89, 912kb
- b. גודל הקובץ המקסימלי הוא 66, 144kb
- c. גודל הקובץ המקסימלי הוא 1TB
- d. גודל הקובץ המקסימלי הוא 4TB

3. פונקציית `wait(&status)`. מה נשמר ב-`status`? סמנו את התשובה הנכונה.

- a. ערך החזרה (return value) של הקוד שהילד ביצע לפני סיום תפקידו.
- b. מצב תהליך הילד (ריצה, מוכן לריצה או חסום).
- c. מצב תהליך ההורה (ריצה, מוכן לריצה או חסום).
- d. ב-`status` נשמר ה-pid של הילד.

4. טבלת דפים היררכית (2 רמות).

- a. יש צורך ב-2 גישות לזיכרון.
- b. יש צורך ב-3 גישות לזיכרון.
- c. הרמה השנייה תכיל את כל המצביעים.
- d. הרמה הראשונה תכיל רק את המצביעים שבשימוש.

5. פסיקת דף (page fault).

- a. אם דף לא נמצא בזיכרון הפיסי אז ייתכן שהוא לא במרחב הלוגי של התכנית או שהוא בדיסק ועוד לא נטען לזיכרון.
- b. בטיפול בפסיקה, מערכת ההפעלה תבדוק ב-PCB של התהליך את רשימת אזורי הזיכרון ולפי זה תדע האם זו שגיאת זיכרון או שצריך להביא את הדף החסר.
- c. לאחר הבאת הדף החסר, יש צורך להפעיל את הפקודה שגרמה לפסיקה מחדש.
- d. כל התשובות נכונות.

6. ניהול רשימת הבלוקים הפנויים (ללא מצביעים / עם מצביעים) - פיצול ואיחוד של בלוקים.

- a. בהקצאה נאחד את הבלוקים כדי למנוע שבירה פנימית ובשחרור נפצל את הבלוק לשני בלוקים כדי למנוע שבירה חיצונית.
- b. בהקצאה נפצל את הבלוק לשני בלוקים כדי למנוע שבירה פנימית ובשחרור נאחד את הבלוקים כדי למנוע שבירה חיצונית.
- c. בהקצאה נפצל את הבלוק לשני בלוקים כדי למנוע שבירה חיצונית ובשחרור נאחד את הבלוקים כדי למנוע שבירה פנימית.
- d. בהקצאה נאחד את הבלוקים כדי למנוע שבירה חיצונית ובשחרור נפצל את הבלוק לשני בלוקים כדי למנוע שבירה פנימית.

7. נניח והחומרה (MMU) פונה ל-TLB בשביל לקבל דף כלשהו. ה-TLB מצא (hit) תרגום עבור הכתובת הלוגית

לכתובת הפיזית. מה השלב הבא בתהליך לקבלת הדף הפיסי?

- a. מעדכנים אותו ב-Parity block.
- b. עוברים לטבלת הדפים כדי לקבל את הכתובת לזיכרון הראשי.
- c. עוברים לזיכרון הראשי ומחפשים את הדף לפי הכתובת.
- d. ניגשים למטמון (Cache) ובמידה וקיים (hit) נשלח אותו ל-CPU.

8. נתון קטע קוד. נניח שמטמון במיפוי לקבוצה מכיל 16 קבוצות, בכל קבוצה 2 בלוקים, כל בלוק בגודל 16 בתים.

גודל המטמון 512 בתים.

```
1 int func() {
2     int x[128], y[128];
3     int i, sum = 0;
4     for (i = 0; i < 128, i++) {
5         sum *= x[i] + y[i];
6     }
7     return sum;
8 }
```

המערך x תופס $128 \cdot 4 = 512$ בתים ומתחיל בכתובת 0.

המערך y תופס $128 \cdot 4 = 512$ בתים ומתחיל מיד אחרי x.

קיים פגיעה של 75%, איזה פתרון יכול לשפר את הפגיעה?

- a. הגדלת המטמון.
- b. הגדלת הבלוק.
- c. הגדלת מספר הקבוצות.
- d. לא ניתן לשפר יותר מ-75% (cold misses).

9. MBR. ה-MBR מכיל:

- a.** מצביעים לבלוקים של הדיסק
- b.** קוד אתחול בלבד
- c.** קוד אתחול וטבלת מחיצות של הדיסק
- d.** טבלת מחיצות של הדיסק בלבד

10. החלפת תהליך (Context Switch)

- a.** בזמן החלפת תהליך, צריך לשמור את האוגרים של התהליך המוחלף, בקובץ.
- b.** בזמן החלפת תהליך, צריך לשמור את הקבצים שהתהליך המוחלף פתח, בדיסק.
- c.** בזמן החלפת תהליך, צריך לשמור את האוגרים של התהליך המוחלף, ב-PCB.
- d.** בזמן החלפת תהליך, צריך לשמור את הקבצים שהתהליך המוחלף פתח, ב-PCB.

מבחן מספר 11

1. פסיקות (Interrupts)

- a.** ניסיון לבצע פקודה לא חוקית יגרום לפסיקה פנימית, חלוקה באפס תגרום לפסיקה פנימית.
- b.** חלוקה באפס תגרום לפסיקה פנימית, פסיקת שעון תגרום לפסיקה פנימית.
- c.** פסיקת שעון תגרום לפסיקה פנימית, חלוקה באפס תגרום לפסיקה חיצונית.
- d.** ניסיון לבצע פקודה לא חוקית יגרום לפסיקה פנימית, חלוקה באפס תגרום לפסיקה חיצונית.

2. יצירת תהליך (fork()). בתיקה "home/user/mydocs/" ישנם שלשה קבצים a, b ו-c. מהו הפלט של התכנית הבאה?

```
int value = 10;
int main() {
    pid_t pid;
    pid = fork();
    if (pid == 0) {
        value += 10;
        execlp("/bin/ls", "ls", "/home/user/mydocs", NULL);
        printf("%d\n", value);
        return 0;
    }
    else if (pid > 0) {
        wait(NULL);
        printf("%d\n", value);
        return 0;
    }
}
```

a. a, b, c
20

b. a, b, c
10

c. a, b, c
10
20

d. 10
a, b, c

3. הפנית קלט/פלט (redirection). הפקודה `ls > abc`.
- a. מנתבת את הפלט של הפקודה `ls` לקלט של הפקודה `abc`.
 - b. כותבת את הפלט של הפקודה `ls` לקובץ `abc`.
 - c. מנתבת את הפלט של הפקודה `abc` לקלט של הפקודה `ls`.
 - d. כותבת את הפלט של הפקודה `ls` לסוף הקובץ `abc`.

4. פתיחת קובץ.
- a. הערך המוחזר מ-`open` הוא אינדקס בטבלת הקבצים הפתוחים.
הערך המוחזר מ-`read` הוא מספר התווים שנקראו.
 - b. הערך המוחזר מ-`open` הוא מצביע למערך התווים שישמשו לקריאה.
הערך המוחזר מ-`read` הוא 0 או 1.
 - c. הערך המוחזר מ-`open` הוא אינדקס בטבלת הקבצים הפתוחים.
הערך המוחזר מ-`read` הוא 0 או 1.
 - d. הערך המוחזר מ-`open` הוא מצביע למערך התווים שישמשו לקריאה.
הערך המוחזר מ-`read` הוא מספר התווים שנקראו.

5. יצירת תהליך (`fork()`). הקובץ `myfile.txt` מכיל שורה אחת: "Hello world". מהו הפלט של התוכנית הבאה?

```
int value = 10;
int main()
{
    pid_t pid;
    pid = fork();
    if (pid == 0) {
        value -= 5;
        execlp("/bin/cat", "cat", "myfile.txt", NULL);
        printf("%d\n", value);
    }
    return 0;
    else if (pid > 0) {
        wait(NULL);
        printf("%d\n", value);
        return 0;
    }
}
```

- a. Hello world
5
- b. Hello world
5
10
- c. 10
10
Hello world
5
- d. Hello world
10

6. `lseek` . פתחנו קובץ בגודל 100 בתים וקראנו ממנו 18 בתים, לאחר מכן ביצענו את הפקודה:

```
pos = lseek(fd, 10, SEEK_CUR);
```

מה הערך שיכנס ל-`pos`?

a. 28

b. 10

c. 9

d. 27

7. ה-`chmod`.

לקובץ `myfile` יש את ההרשאות הבאות:

```
rw-rw-rw-
```

איזו פקודת `chmod` תשנה את ההרשאות ל:

```
rwxrw-rw-
```

a. `chmod 744 myfile`

b. `chmod 755 myfile`

c. `chmod 766 myfile`

d. `chmod 733 myfile`

8. שימוש ב-`umask`.

כשיוצרים קובץ באמצעות הפקודה `touch`, הפקודה יוצרת אותו עם ההרשאות:

```
rw-rw-rw-
```

איזה `umask` יגרום לקובץ להיווצר עם ההרשאות:

```
rw-----
```

a. 400

b. 600

c. 77

d. 33

9. מצבי החומרה (CPU mode)

a. מעבר ממצב רגיל (User) למצב מיוחס (Privileged), נובע מקריאת מערכת ((`System Call`), מפסיקה

פנימית אך לא מפסיקה חיצונית.

b. מעבר ממצב רגיל (User) למצב מיוחס (Privileged), נובע מקריאת מערכת ((`System Call`), מפסיקה

פנימית ומפסיקה חיצונית.

c. מעבר ממצב רגיל (User) למצב מיוחס (Privileged), נובע מקריאת מערכת ((`System Call`), מפסיקה

חיצונית אך לא מפסיקה פנימית.

d. מעבר ממצב רגיל (User) למצב מיוחס (Privileged), נובע מפסיקה פנימית, מפסיקה חיצונית אך לא

מקריאת מערכת ((`System Call`).

10. קריאת מערכת (System Call)

- a.** קריאת מערכת תעביר את המחשב למצב מיוחס.
קריאה לפונקציה לא תעביר את המחשב למצב מיוחס.
- b.** קריאת מערכת לא תעביר את המחשב למצב מיוחס.
קריאה לפונקציה לא תעביר את המחשב למצב מיוחס.
- c.** קריאת מערכת תעביר את המחשב למצב מיוחס.
קריאה לפונקציה תעביר את המחשב למצב מיוחס.
- d.** קריאת מערכת לא תעביר את המחשב למצב מיוחס.
קריאה לפונקציה תעביר את המחשב למצב מיוחס.

מבחן מספר 12

1. ה-PCB.

- a. כאשר מבצעים fork() אז parent process מועתק מההורה לילד אך open file descriptors לא מועתק.
- b. כאשר מבצעים fork() אז parent process לא מועתק מההורה לילד וגם open file descriptors לא מועתק.
- c. כאשר מבצעים fork() אז parent process מועתק מההורה לילד וגם open file descriptors מועתק.
- d. כאשר מבצעים fork() אז parent process לא מועתק מההורה לילד אך open file descriptors מועתק.

2. יצירת תהליך (fork()). מהו הפלט של התכנית הבאה?

```
int main () {  
    printf ("A\n");  
    if (fork() > 0)  
        printf ("B\n");  
    exit(0);  
}
```

a.

B
B
A

יא

B
A
B

b.

A
B

יא

B
A

.c

A

B

B

IX

B

A

B

.d

A

A

B

IX

A

B

A

3. הפנית קלט/פלט (redirection). הפקודה xyz | abc

a. מנתבת את הפלט של הפקודה abc לקלט של הפקודה xyz.

b. כותבת את הפלט של הפקודה abc לקובץ xyz.

c. מנתבת את הפלט של הפקודה xyz לקלט של הפקודה abc.

d. קוראת את הקלט של הפקודה abc מתוך הקובץ xyz.

4. שימוש ב-lseek. פתחנו קובץ שגודלו 1000 בתים, לאחר מכן ביצענו את הפקודות:

```
lseek(fd, 0, SEEK_END);  
lseek(fd, 100, SEEK_SET);  
pos = lseek(fd, 0, SEEK_CUR);
```

מה הערך שיכנס ל-pos?

a. 99

b. 1099

c. 100

d. 1000

5. שימוש ב-chmod.

לקובץ myfile יש את ההרשאות הבאות:

rwXrwxr--

איזו פקודת chmod תשנה את ההרשאות ל:

rwXrw-rw-

a. chmod u-x,g+w myfile

b. chmod u-w,g+x myfile

c. chmod g-x,o+w myfile

d. chmod g-w,u+x myfile

6. קישורים. אם הקובץ b הוא symbolic link לקובץ a והקובץ c הוא hard link לקובץ a:

a. לקבצים b, a - ו- c יהיה אותו מספר .inode

b. לקבצים a - ו- b יהיה אותו מספר .inode

c. לקבצים a - ו- c יהיה אותו מספר .inode

d. לקבצים b - ו- c יהיה אותו מספר .inode

7. הרשאות לקבצים (File Permissions)

הקובץ file נמצא בתוך התיקיה dir שנמצא בתוך תיקיית השורש, כך: dir/file/

מהן ההרשאות הדרושות כדי לכתוב לקובץ file שבתיקיה dir?

a. הרשאת ביצוע לתיקיית השורש ולתיקיה dir, וכן הרשאת כתיבה ל- file.

b. הרשאת ביצוע לתיקיית השורש ולתיקיה dir, וכן הרשאת כתיבה ל- dir והרשאת כתיבה ל- file.

c. הרשאת כתיבה לתיקיית השורש ולתיקיה dir, וכן הרשאת כתיבה ל- file.

d. הרשאת קריאה לתיקיית השורש ולתיקיה dir, וכן הרשאת כתיבה ל- file.

8. שימוש ב-fork. כמה פעמים תדפיס התוכנית הבאה "I have finished"?

```
#include <stdio.h>
main() {
    for (i = 0 ; i < 3 ; i++)
        fork ();
    printf ("I have finished.\n");
}
```

a. 5

b. 8

c. 6

d. 4

9. כאשר מחברים דיסקים בשיטת RAID level 5:

a. קריאת קובץ תהיה יותר מהירה, אך אם דיסק אחד קורס המידע הולך לאיבוד.

b. קריאת קובץ תהיה יותר מהירה, אם דיסק אחד קורס המידע לא הולך לאיבוד.

c. קריאת קובץ תהיה יותר איטית, אך אם דיסק אחד קורס המידע לא הולך לאיבוד.

d. קריאת קובץ תהיה יותר איטית, אם דיסק אחד קורס המידע הולך לאיבוד.

10. שיטות להקצאת קבצים (File Allocation Methods)

- a.** בהקצאה משורשרת (Chained Allocation) יש יתרון כי קריאה של קובץ היא יותר מהירה, ויש חיסרון כי קשה לנצל את כל הבלוקים בדיסק.
- b.** בהקצאה משורשרת (Chained Allocation) יש יתרון כי יש ניצול של כל הבלוקים בדיסק, ויש חיסרון כי קריאה של קובץ היא פחות מהירה.
- c.** בהקצאה משורשרת (Chained Allocation) יש יתרונות כי יש ניצול של כל הבלוקים בדיסק וקריאה של קובץ היא יותר מהירה.
- d.** בהקצאה משורשרת (Chained Allocation) יש חסרונות כי קריאה של קובץ היא פחות מהירה וקשה לנצל את כל הבלוקים בדיסק.

11. יצירת תהליך. בתיקיה "home/user/mydocs/" ישנם שלשה קבצים a, b ו-c.

מהו הפלט של התוכנית הבאה?

```
int main() {  
    int value = 10;  
    pid_t pid;  
    pid = fork();  
    if (pid == 0) {  
        value += 20;  
        printf("%d\n", value);  
        execlp("/bin/ls", "ls", "/home/user/mydocs", NULL);  
        printf("%d\n", value);  
        return 0;  
    }  
    else if (pid > 0) {  
        value += 30;  
        wait(NULL);  
        printf("%d\n", value);  
        return 0;  
    }  
}
```

a. 30

a, b, c

40

b. 30

a, b, c

30

40

c. 30

a, b, c

60

d. 30

a, b, c

30

60

תשובות

מבחן 6	
d	1
b	2
a	3
a	4
a	5
a	6
b	7
b	8
c	9
c	10

מבחן 5	
a	1
d	2
d	3
c	4
c	5
b	6
b	7
b	8
d	9
a	10

מבחן 4	
d	1
a	2
b	3
a	4
d	5
c	6
d	7
b	8
b	9
b	10

מבחן 3	
b	1
d	2
c	3
d	4
a	5
a	6
a	7
a	8
d	9
c	10

מבחן 2	
a	1
c	2
a	3
c	4
c	5
b	6
a	7
c	8
b	9
c	10

מבחן 1	
c	1
a	2
c	3
b	4
b	5
b	6
b	7
a	8
a	9
a	10

מבחן 12	
d	1
b	2
a	3
c	4
c	5
c	6
a	7
b	8
b	9
b	10
a	11

מבחן 11	
a	1
b	2
b	3
a	4
d	5
a	6
c	7
c	8
b	9
a	10

מבחן 10	
c	1
d	2
a	3
b	4
d	5
b	6
d	7
b	8
c	9
c	10

מבחן 9	
b	1
c	2
d	3
c	4
d	5
d	6
c	7
a	8
c	9
c	10

מבחן 8	
d	1
b	2
c	3
a	4
a	5
b	6
a	7
d	8
c	9
d	10

מבחן 7	
b	1
a	2
b	3
d	4
b	5
c	6
c	7
d	8
b	9
a	10

RAID <i>D = Disks, B = Blocks</i>				
	תאור	ביצועים	נפח	אמינות
RAID-0 פסים	מחלקים את הדיסקים לפסים. כל בלוק בדיסק אחר.	הטוב ביותר - ניתן לגשת לדיסקים במקביל.	טוב מאוד - $D \cdot B$	גרוע - אין יתירות כלל.
RAID-1 שכפול	כל הנתונים נכתבים לשני דיסקים (ראי).	טוב - ניתן לגשת ל-2 הדיסקים במקביל.	סביר: $\frac{D \cdot B}{2}$	טוב מאוד - כאשר כונן נכשל - ניגש לשני.
RAID-4 זוגיות	בדיסק האחרון נשמר מידע לשחזור נתונים במקרה של תקלה בדיסק אחר.	טוב - ניתן לגשת לדיסקים במקביל.	טוב מ-RAID-1: $(D - 1) \cdot B$	טוב - נשתמש ב-parity עם XOR לשחזור הדיסק.
RAID-5 זוג בסבב	הבלוקים של הזוגיות מפוזרים בסבב על כל הדיסקים.	טוב יותר מ-RAID-4 כיוון שמונע צוואר בקבוק	טוב מ-RAID-1: $(D - 1) \cdot B$	טוב - נשתמש ב-parity עם XOR לשחזור הדיסק.

הקצאות של בלוקים לקובץ כל קובץ מורכב מאוסף של בלוקים			
	תיאור	יתרונות	חסרונות
הקצאה רציפה	כל קובץ נשמר כרצף של בלוקים.	<ul style="list-style-type: none"> קריאה וכתיבה מהירה. במדריך יש רק את מספר הבלוק הראשון ואת מספר הבלוקים בקובץ. דילוג מהיר בין הבלוקים. 	<ul style="list-style-type: none"> שבירה חיצונית. לדעת מראש את גודל הקובץ. קשה לנצל את כל הבלוקים בדיסק.
הקצאה משורשרת	כל קובץ הוא רשימה מקושרת של בלוקים. בכל בלוק, הבתים הראשונים הם מצביע לבלוק הבא.	<ul style="list-style-type: none"> אין שבירה חיצונית. מספיק לשמור את הבלוק הראשון והאחרון (או מספר הבלוקים). קובץ גדל כל עוד יש בלוקים פנויים בדיסק. 	<ul style="list-style-type: none"> גישה איטית (לעבור על הכל). המצביע תופס חלק מהבלוק.
הקצאה משורשרת עם FAT	המצביעים לא בבלוקים אלא בטבלה נפרדת שנשמרת בראש הדיסק.	<ul style="list-style-type: none"> גישה אקראית מהירה יותר מהקצאה משורשרת רגילה. 	<ul style="list-style-type: none"> אם הדיסק גדול אז הטבלה תתפוס הרבה מקום בזיכרון.
הקצאה עם מצביעים	כל המצביעים לבלוקים של הקובץ נמצאים בבלוק אחד בקובץ. משומש בלינוקס.	<ul style="list-style-type: none"> אין שבירה חיצונית. ניצול כל הבלוקים בדיסק. קובץ גדל כל עוד יש בלוקים פנויים בדיסק. גישה אקראית מהירה. אפשר להקצות מצביעים לבלוק של מצביעים (single,double,triple). 	<ul style="list-style-type: none"> גישה איטית בהשוואה להקצאה רציפה. טבלת המצביעים תופסת מקום בזיכרון.

חלאס.