

בחינה

פקולטה : מדעי הטבע
מחלקה : מדעי המחשב
שם הקורס : תכנות מתקדם
קוד הקורס : 7031010
תאריך הבחינה : 7/2/2022
סמסטר : א, מועד : 2
משך הבחינה : שעתיים
שם המרצה : ד"ר פנחס ויסברג

לפניכם 10 שאלות, משקל כל שאלה 10 נקודות
יש לענות על כל השאלות
ללא חומר עזר, אפשר להשתמש במחשבון

בהצלחה

1. fopen (10)

הסבירו בפרוט מדוע קריאה וכתיבה של קובץ שנפתח עם fopen יהיו בדרך כלל מהירות יותר מקריאה וכתיבה של קובץ שנפתח עם open ?

תשובה (מתוך השקפים) :

פונקציות הקריאה והכתיבה של הספרייה הסטנדרטית משתמשות ב- buffer (4K) כדי להפחית את מספר הקריאות למערכת.
בקריאה מדיסק, בפעם הראשונה ובכל פעם שה- buffer ריק, מתבצעת קריאה בגודל ה- buffer, האפליקציה קוראת מה- buffer תווים עד שמתרוקן.
בכתיבה לדיסק, האפליקציה כותבת ל- buffer עד שמתמלא, ואז הספרייה כותבת לדיסק.

2. (10) copy on write

א. מה הבעיה ש-copy on write פותרת.

ב. הסבירו בפרוט איך פועל הפתרון?

תשובה (מתוך השקפים):

א. fork יוצר תהליך ילד שבו קטעי הזיכרון הם העתק של קטעי הזיכרון של ההורה. בדרך כלל הילד מבצע exec מיד אחרי ה-fork ומחליף את קטעי הזיכרון, אם כן ביצוע ההעתקה היה מיותר.

ב. במקום להעתיק את קטעי הזיכרון אפשר לשתף אותם בין ההורה לילד, לסמן אותם read-only בטבלאות הדפים של ההורה והילד ולזכור שהם copy-on-write.

3. (10) lseek

מה יכיל הקובץ "file.txt" לאחר כל אחת מארבע פקודות write() שבתכנית.

```
int main(void) {
    char    buf1[] = "123456789012";
    char    buf2[] = "AA";
    int     fd;
    fd = open("file.txt", O_WRONLY | O_TRUNC | O_CREAT
               , S_IRUSR | S_IWUSR);
    write(fd, buf1, 12);
    lseek(fd, -3, SEEK_END);
    write(fd, buf2, 2);
    lseek(fd, -5, SEEK_CUR);
    write(fd, buf2, 2);
    lseek(fd, 2, SEEK_SET);
    write(fd, buf2, 2);
}
```

תשובה:

```
123456789012
123456789AA2
123456AA9AA2
12AA56AA9AA2
```

4. (10) דיסק

לדיסק יש את הנתונים הבאים:

זמן חיפוש (seek) : 8ms
מהירות הסיבוב: 3000RPM (סיבובים לדקה)
מספר הסקטורים במסילה: 250

כמה זמן ייקח לקרוא סקטור אחד ?

תשובה:

$$T_{\text{full_rotation}} = 1/3,000 * 60 * 1000 = 20 \text{ ms}$$

$$T_{\text{rotation}} = T_{\text{full_rotation}} * 0.5 = 10 \text{ ms}$$

$$T_{\text{transfer}} = T_{\text{full_rotation}} / 250 \text{ sectors} = 0.08 \text{ ms}$$

זמן הגישה:

$$T_{\text{seek}} + T_{\text{rotation}} + T_{\text{transfer}} = 8 + 10 + 0.08 = 18.08$$

5. (10) ניהול קבצים פתוחים

קבצים פתוחים מנוהלים על ידי הטבלאות:

Descriptor table - לכל תהליך יש טבלה כזו, הטבלה מכילה את הקבצים שהתהליך פתח.

כל קובץ פתוח מצביע לאובייקט של הטבלה הבאה:

Open file table - אובייקט בטבלה זו מכיל פרטים אודות הקובץ הפתוח.

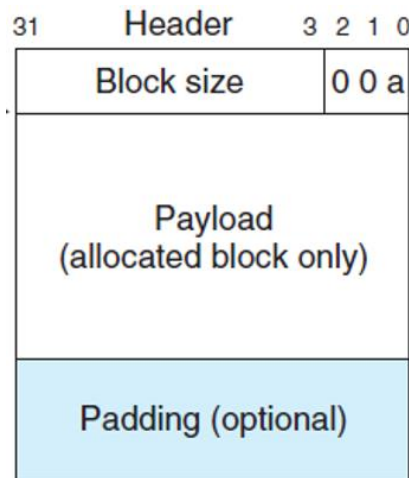
מהם שתי האפשרויות ששתי שורות שונות ב- Descriptor table יצביעו על אותה שורה ב- Open file table.

תשובה:

א. dup()
ב. fork()

6. malloc (10)

בניהול זיכרון דינמי בשיטת Implicit Free List (ללא מצביעים מפורשים) מבנה הבלוק הוא:



עבור שתי הפעולות שלהלן הסבירו האם הן מתבצעות

- בזמן קבוע
- או בזמן לינארי לפי מספר הבלוקים בערימה
- או בזמן לינארי לפי מספר הבלוקים הפנויים בערימה

- א. (5) הקצאת בלוק.
ב. (5) מיזוג בלוק עם הבלוק שנמצא בזיכרון בכתובת נמוכה ממנו.

תשובה :

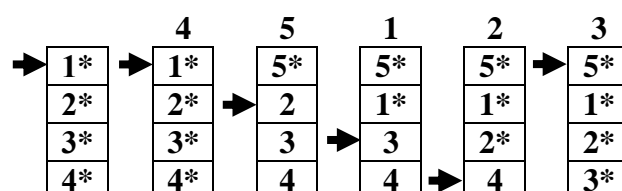
- א. בזמן לינארי לפי מספר הבלוקים בערימה
ב. בזמן לינארי לפי מספר הבלוקים בערימה

7. (10) אלגוריתם השעון

נניח שישנם 4 דפים פיסיים שמכילים את ארבעת הדפים הלוגיים 1,2,3,4. כוכבית מימין למספר הדף מסמנת שהייתה גישה לדף (הביט used דלוק). חץ משמאל למספר הדף מסמן שהמחוג מצביע עליו. (המחוג נע כלפי מטה, כשמגיע למטה חוזר בקפיצה למעלה)

התכנית המשיכה לגשת לסדרת הדפים (משמאל לימין):
4,5,1,2,3

ציירו והשלימו איך תראה כל עמודה לאחר הבאת הדף החסר:



8. FAT (10)

מהם שני יתרונות בהקצאה משורשרת בשיטת FAT לעומת הקצאה משורשרת רגילה.

תשובה :

- המצביעים לא נמצאים בתוך הבלוקים אלא בטבלה נפרדת שנשמרת בתחילת הדיסק, הטבלה מועתקת לזיכרון, לכן :
- א. גישה אקראית תהיה יותר מהירה כי השרשרת נמצאת בזיכרון ולא בבלוקים של הדיסק.
- ב. הבלוקים יהיו בגודל 512 (מתאים לגדלים אחרים במערכת ההפעלה)

9. inode (10)

נניח שגודל הבלוקים במערכת קבצים הוא 8KB וגודל מצביע לבלוק הוא 4 בתים. inode מכיל 12 מצביעים ישירים (לבלוקים של data), ומכיל מצביע אחד עקיף (לבלוק של מצביעים). מה הגודל המקסימלי (בבתים) של קובץ במערכת קבצים זו ?

תשובה :

מספר המצביעים בבלוק הוא :
 $8192 / 4 = 2048$

אם כן גודל הקובץ הוא :
 $(12 + 2048) * 8KB = 16,480KB$

10. (10) טבלת דפים היררכית

טבלת דפים מכילה מצביעים (כתובות) של דפים פיסיים. טבלת דפים היררכית (שתי רמות) מכילה בלוק של מצביעים לבלוקים של מצביעים לדפים פיסיים. מספר המצביעים (השורות) בבלוק של הרמה העליונה שווה למספר המצביעים (השורות) בכל אחד מהבלוקים שברמה מתחתיה.

א. מה היתרון של טבלת דפים היררכית?

ב. אם גודל הכתובת 21 ביטים וגודל הדף 2K, כמה מצביעים (שורות) יש בכל בלוק?

תשובה :

- א. תופסת פחות מקום.
- ב. כדי לבנות את טבלת הדפים בשתי רמות, נחלק את הכתובת לשלושה חלקים :

page number		page offset
p_1	p_2	d

מאחר שגודל הדף הוא $2K = 2^{11}$ אם כן d הוא 11 ביטים ו- p_1 ו- p_2 הם 5 ביטים כל אחד. מספר השורות בכל בלוק הוא $2^5 = 32$