עבודה 2 מבני נתונים

אתחול המערכת:

ראשית מתבצעת קריאה לבנאי אשר מאתחל רשימה מקושרת ריקה ושני מערכים ריקים בגודל n - פעולות בזמן קבוע.

בנוסף קורא הבנאי לשתי פונקציות עזר:

1. initiateSec: פונקציה זו נועדה לאתחל את הזיכרון המשני במחרוזות ריקות. הזיכרון הכולל שנדרש הינו m.

2. initiateMain: פונקציה זו נועדה לאתחל את הזיכרון הראשי, הפונקציה יוצרת n דפים חדשים שתופסים בזיכרון n*page size מקום. בנוסף במערך הנועד לשמור מצביעים יתווספו מצביעים לדפים באינדקסים הרלוונטיים – פעולה זו אינה יוצרת דפים חדשים אלא רק מצביעה כלומר מייצרת מסלול לדפים קיימים ולכן עבורה לא נצטרך לפנות מקום חדש בזיכרון.

סך כל הזיכרון הנדרש הינו n*page size + 2m + c כלומר נקבל סיבוכיות זיכרון $O(n * pageSize + m)$

זמני ריצה של פעולות המערכת:

בכל קריאה וכתיבה, נעזר בפונקציה check שהיא זו שעושה את הפעולות הנדרשות טרם הקריאה או הכתיבה. פונקציה זו בודקת האם הדף הדרוש נמצא כבר בזיכרון הראשי ובהתאם לכך ולשיטת העבודה (FIFO or LRU) מבצעת פעולות בנתונים הרלוונטיים.

FIFO:

Operator:

1. תחילה נבדוק האם האיבר קיים בזיכרון הראשי, פעולה המתבצעת ע"י השוואה של איבר במערך.
   1.1. אם הדף הדרוש נמצא בזיכרון הראשי: נמשוך את הדף הדרוש ונחזיר אותו.
   1.2. אחרת: ניצור דף חדש (פעולה זו לוקחת זמן קבוע)
      a. נמשוך את הדף שעתיד לצאת מהזיכרון הראשי, יסומן בtail.
      b. נכניס את התוכן שעודכן ב tail במקומו בזיכרון המשני.
      c. נעדכן את מערך העזר שדף זה אינו עוד בזיכרון הראשי.
      d. נעדכן את מערך העזר שנכנס דף חדש לזיכרון הראשי.
      e. נכניס את הדף החדש לזיכרון הראשי (הכנסה בזמן קבוע).
      f. נחזיר את הדף החדש לפונקציה.

LRU:

Operator:

2. תחילה נבדוק האם האיבר קיים בזיכרון הראשי, פעולה המתבצעת ע"י השוואה של איבר במערך.
   2.1. אם הדף הדרוש נמצא בזיכרון הראשי: נמשוך את הדף הדרוש אנחנו הולכים לעבוד, נקרא לו currPage. ונפריד לשני מקרים:
      a. נבדוק אם currPage הוא הבא לצאת מהרשימה.
         1. נוציא אותו (זמן קבוע).
         2. נחזיר אותו (זמן קבוע).
      b. נבדוק אם currPage הוא לא בראש הרשימה.
         1. נגדיר את העוקב של currPage להיות העוקב של הקודם של currPage.

2. נגדיר את הקודם של currPage להיות הקודם של העוקב של currPage.
3. נכניס את currPage לראש הרשימה.

c. נחזיר את currPage. נוסיף ונאמר כי אם currPage הוא לא מקיים את מקרים a,b אז הוא בראש הרשימה ולכן רק צריך להחזיר אותו.

2.2. אחרת: ניצור דף חדש (פעולה זו לוקחת זמן קבוע).

g. נמשוך את הדף שעתיד לצאת מהזיכרון הראשי, יסומן בtail.
h. נכניס את התוכן שעודכן ב tail במקומו בזיכרון המשני.
i. נעדכן את מערך העזר שדף זה אינו עוד בזיכרון הראשי.
j. נעדכן את מערך העזר שנכנס דף חדש לזיכרון הראשי.
k. נכניס את הדף החדש לזיכרון הראשי (הכנסה בזמן קבוע).
l. נחזיר את הדף החדש לפונקציה.

- אנו יודעים שההסבר מסורבל מעט אך סך הכל משתמשים במצביעים על מנת לעקוף דף כלשהו או לסמן דף כלשהו בתפקיד מסוים (tail or head).

פונקציית READ:

השיטה הזו משתמש בשיטת העזר OPERATOR ולכן מבצעת פעולה אחת נוספת של החזרת הדף (פעולה בזמן קבוע).

פונקציית WRITE:

השיטה הזו משתמש בשיטת העזר OPERATOR ולכן מבצעת פעולה אחת נוספת של שינוי המחרוזת בדף (פעולה בזמן קבוע).

- שמנו לב שהבדיקות הדרושות בקריאה וכתיבה של דף הן שונות במעט בלבד ולכן החלטנו ליצור פונקציית עזר על מנת להימנע מכפל קוד. מכאן שהשיטות READ ו WRITE עצמן לא מבצעות פעולות כבדות נוספות וזמן הריצה שלהם תלוי ברובו המוחלט בשיטת העזר.