

שיעורי בית 5, מערכות הפעלה 2025

- תאריך הגשה: 31.1
- בשאלות קוד: אין להשתמש בספריות מוכנות (לדוגמה: ספרייה שמממשת LRU **אסורה!**).

שאלה 1:

- במערכת זיכרון מסוימת, לכל דף יש מספר מזהה i . מכיוון שהלוקליות בגישה לזיכרון הינה גדולה מאוד, הגדירו את מדיניות הבאה:
- אם נרצה להביא לזיכרון דף מספר i , והזיכרון מלא, נפנה מהזיכרון דף j כך ש $|i - j|$ הינו מקסימלי.
- א. הנח שגודל הטבלה הינה 3 דפים. עבור סדרת הגישות הבאה, רשום את: תוכן הזיכרון וכמה PF מתבצעים במהלך הריצה?
- $[1, 2, 3, 4, 3, 2, 1, 4, 1, 3]$: יש לקרוא משמאל לימין.
- ב. תן דוגמה לסדרת גישות (אחרת) שבה האלגוריתם LRU יעיל יותר מהאלגוריתם שהוצג מקודם.
- ג. האם LRU תמיד יותר טוב מהאלגוריתם הנתון? הוכח / הבא דוגמה נגדית.

שאלה 2:

במערכת מחשב התומכת ב- multi-level-paging נתון כי גודלו של כל דף במערכת הוא 64 KB (kilobytes 64), גודלה של כל טבלה בכל רמה הוא דף יחיד וגודלה של כל כניסה בכל אחת מן הטבלאות (entry table page) הוא 4B (4Bytes) מה רמות נצטרך בטבלאות הדפים כדי למפות 16TB? הסבר בפירוט.

שאלה 3: (זוהי שאלה 4 מדף עבודה מס 3)

- A. Consider the following series of accesses to memory pages: 0,1,2,3,0,1,4,0, 1,2,3,4. How many page faults would happen, when using FIFO replacement policy and a memory which contains only 3 frames?
- B. Repeat A, when given that the memory contains 4 frames.
- C. Prove or disprove: When using FIFO algorithm, the set of pages which reside in the memory of n pages is always a subset of the set of pages which reside in the memory of $n+1$ pages.
- D. Prove or disprove: When using FIFO algorithm, enlarging the memory may increase the number of page faults.
- E. Prove or disprove: When using LRU algorithm, the set of pages which reside in the memory of n pages is always a subset of the set of pages which reside in the memory of $n+1$ pages.
- F. Prove or disprove: When using LRU algorithm, enlarging the memory may increase the number of page faults.

שאלה 4:

הוכח/ הפרך- בסדרת גישות לדפים ניגשים לכל דף לכל היותר שלוש פעמים. מספר page faults שגורם שימוש ב $FIFO 2^{nd} chance$ תמיד שווה למספר page faults שגורם שימוש ב-LRU.

שאלה 5:

תכנית קוראת קובץ בשם /usr/tmp/file.txt במערכת קבצים Unix. גודל הקובץ הוא 10.5 KB. הניחו שגודלו של כל בלוק במערכת הקבצים הוא 1 KB ושבתחילת ריצת התכנית מבני הנתונים היחידים שנמצאים בזיכרון ה-RAM הינם root directory וה-i-node שלו. הניחו גם שכל inode נמצא

בשלמותו בבולוק יחיד של הדיסק ושגם כל תיקיה (directory) נמצאת בשלמותה בבולוק יחיד של הדיסק. הניחו בנוסף שה-buffer cache גדול וכל בלוק המובא ל-RAM במהלך ריצה נשאר בו עד סוף הריצה ושכל inode יש 10 שדות של direct block numbers. כמה בלוקים תצטרך התכנית לקרוא מן הדיסק על מנת להביא את כל תוכנו של הקובץ /usr/tmp/file.txt ל-RAM? פרטו את אופן החישוב

שאלה 6- תכנותית:

כתבו פונקציה שמייצרת וקטור מרחקים (distance vector) עבור מנגנון (LRU) Least Recently Used לניהול זיכרון.

הפונקציה מקבלת שני פרמטרים:

1. `table_size` - גודל טבלת הזיכרון הפיזי (מספר מסגרות זמינות).
2. `access_vector` - וקטור המייצג את רצף הגישות לזיכרון (גודל הווקטור אינו ידוע מראש).

הערות:

- **הניחו שגודל הווקטור אינו ידוע מראש ולכן יש להתחשב בכך בעת מימוש.**

בהצלחה

צוות הקורס