

Operating Systems – 234123

Homework Exercise 4 – Dry

Teaching Assistant in charge:

Reda Igbaria

Assignment Subjects & Relevant Course material

Virtual Memory

Lectures 4-5, Recitations 10-11

Submission Format

1. Only **typed** submissions in **PDF** format will be accepted. Scanned handwritten submissions will not be graded.
2. The dry part submission must contain a single PDF file named with your student IDs –
DHW4_123456789_300200100.pdf
3. The submission should contain the following:
 - a. The first page should contain the details about the submitters - Name, ID number and email address.
 - b. Your answers to the dry part questions.
4. Submission is done electronically via the course website, in the **HW4 - Dry** submission box.

Grading

1. **All** question answers must be supplied with a **full explanation**. Most of the weight of your grade sits on your **explanation** and **evident effort**, and not on the absolute correctness of your answer.
2. Remember – your goal is to communicate. Full credit will be given only to correct solutions which are **clearly** described. Convoluting and obtuse descriptions will receive low marks.

Questions & Answers

- The Q&A for the exercise will take place at a public forum Piazza **only**. Please **DO NOT** send questions to the private email addresses of the TAs.
- Critical updates about the HW will be published in **pinned** notes in the piazza forum. These notes are mandatory and it is your responsibility to be updated.

A number of guidelines to use the forum:

- Read previous Q&A carefully before asking the question; repeated questions will probably go without answers
- Be polite, remember that course staff does this as a service for the students
- You're not allowed to post any kind of solution and/or source code in the forum as a hint for other students; In case you feel that you have to discuss such a matter, please come to the reception hour
- When posting questions regarding hw4, put them in the hw4 folder

Late Days

- Please **DO NOT** send postponement requests to the TA responsible for this assignment. Only the **TA in charge** can authorize postponements. In case you need a postponement, please fill out the attached form : <https://forms.gle/G6yyFkZrbgbXxrtY6>

שאלה 1

עדן, זמרת פופולרית, סבלה ממחסור בזיכרון פיזי במחשב שלה (בעל מעבד אינטל 32-ביט וזיכרון פיזי בנפח 4GB) ולכן הציעה תכן חדש של מעבד המרחיב את מרחב הזיכרון הפיזי מ-32 ל-40 ביט. כתוצאה מכך, במימוש של עדן יש שלוש רמות תרגום בטבלת הדפים. שאר נתוני המעבד של עדן זהים לאלו של מעבד IA-32, כלומר נתוני המערכת החדשה הם:

32bit	רוחב כתובת וירטואלית
40bit	רוחב כתובת פיזית
4KB	גודל דף/מסגרת/מגירה
4KB	גודל מסגרת של טבלת דפים (בכל הרמות)
12bit	מספר ביטים לדגלים והרשאות בכל כניסה בטבלת הדפים

1. (5 נק') בהנחה שגודל כניסה בטבלת הדפים **מעוגל למעלה לחזקה שלמה של 2**, מהו אופן חלוקת הכתובת הוירטואלית לשדות בתהליך תרגום כתובות (page walk)?

index3	index2	index1	offset	
2	9	9	12	א.
2	10	10	10	ב.
1	9	10	12	ג.
4	9	9	10	ד.
2	10	10	12	ה.
2	9	9	10	ו.

נימוק:

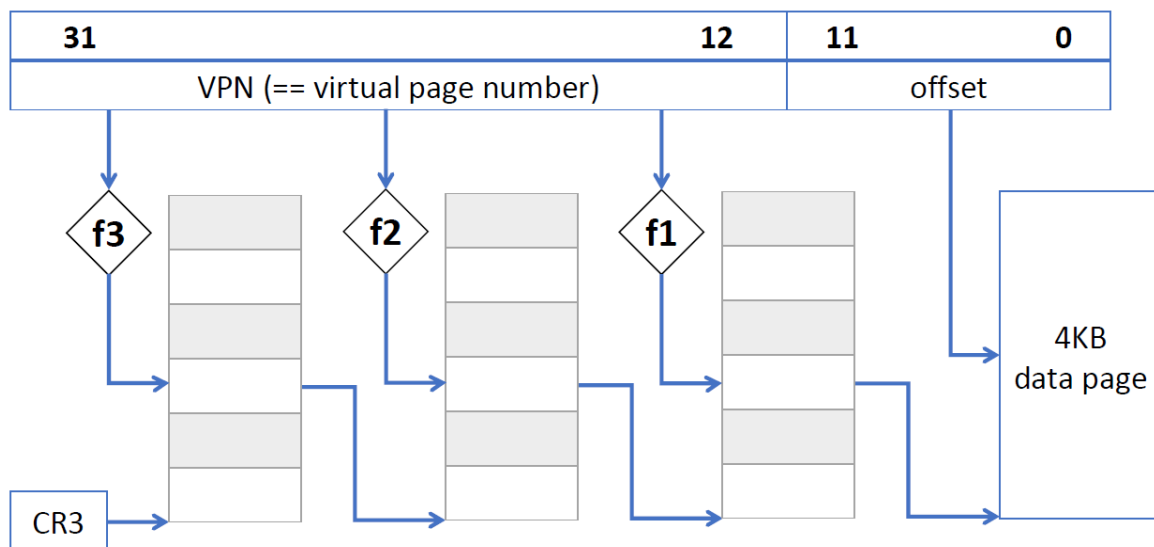
לבעלה של עדן, שוקי, אין שום תואר מהטכניון, ולמרות זאת הוא הבחין כי המימוש של עדן בזבזני בגלל שגודל הכניסות בטבלת הדפים מעוגל למעלה לחזקה שלמה של 2.

2. (5 נק') מהו הגודל המינימלי האפשרי של כניסה בטבלת הדפים אם לא מעגלים אותו למעלה?

- a. 3 בתים
- b. 4 בתים
- c. 5 בתים
- d. 6 בתים
- e. 7 בתים
- f. אף תשובה אינה נכונה

נימוק:

בהמשך לסעיף הקודם, שוקי (בעלה של עדן) הציע מימוש חדש לטבלת הדפים שבו כל כניסה בטבלת הדפים (בכל הרמות) היא בגודל המינימלי מהסעיף הקודם. במימוש של שוקי, כמו במימוש המקורי של טבלת הדפים במעבדי אינטל, דפים סמוכים בזיכרון הוירטואלי נשמרים בכניסות סמוכות בטבלת הדפים. שוקי הבחין שבמימוש החדש הכתובת הוירטואלית אינה מתפרקת לשדות של אינדקסים ויש צורך בחישובים מורכבים על מנת למצוא את האינדקס המתאים בכל טבלה (כלומר בכל רמה בעץ). להלן שרטוט הממחיש את אופן התרגום:



בשרטוט רואים שלוש פונקציות (f1, f2, f3) המקבלות את מספר הדף הוירטואלי VPN ומחזירות, בהתאמה, שלושה אינדקסים לשלושת הרמות בטבלת הדפים. בכל הסעיפים הבאים, הפעולות חלוקה / ומודולו % הן פעולות בשלמים. למשל: $1=1024/819$

$$205 = 1024 \% 819$$

3. (5 נק') מהי הפונקציה f_1 ?

- a. $f_1(vpn) = vpn/819$
- b. $f_1(vpn) = vpn \% 819$
- c. $f_1(vpn) = (vpn/819) \% 819$
- d. $f_1(vpn) = (vpn/819)/819$
- e. $f_1(vpn) = (vpn \% 819)/819$
- f. $f_1(vpn) = ((vpn/819) \% 819)/819$

נימוק:

4. (3 נק') מהי הפונקציה f_2 ?

- a. $f_2(vpn) = vpn/819$
- b. $f_2(vpn) = vpn \% 819$
- c. $f_2(vpn) = (vpn/819) \% 819$
- d. $f_2(vpn) = (vpn/819)/819$
- e. $f_2(vpn) = (vpn \% 819)/819$
- f. $f_2(vpn) = ((vpn/819) \% 819)/819$

נימוק:

5. (2 נק') מהי הפונקציה f_3 ?

- a. $f_3(vpn) = vpn/819$
- b. $f_3(vpn) = vpn \% 819$
- c. $f_3(vpn) = (vpn/819) \% 819$
- d. $f_3(vpn) = (vpn/819)/819$
- e. $f_3(vpn) = (vpn \% 819)/819$
- f. $f_3(vpn) = ((vpn/819) \% 819)/819$

נימוק:

6. (5 נק') מה היתרון של המערכת שהציע שוקי על פני המערכת שהציעה עדן?

- a. מיפוי של מרחב זיכרון וירטואלי גדול יותר.
- b. מיפוי של מרחב זיכרון פיזי גדול יותר.
- c. ה-TLB אפקטיבי יותר בגלל שהוא מכסה יותר זיכרון.
- d. טבלאות הדפים של תהליכי משתמש תופסות נפח קטן יותר בזיכרון.
- e. פחות פרגמנטציה חיצונית, כלומר יותר זיכרון רציף.
- f. אף תשובה אינה נכונה.

נימוק:

שאלה 2

מהי סדרת האירועים אשר מתרחשת כאשר תוכנית ניגשת לכתובת זיכרון וירטואלית ומפעילה את מנגנון copy-on-write?

- a. TLB miss → page table walk → page fault
- b. TLB hit → page table walk → page fault
- c. page fault → page table walk → TLB miss
- d. page fault → page table walk → TLB hit
- e. page table walk → TLB miss → page fault
- g. page table walk → TLB hit → page fault

נמקו:

נתונה התוכנית הבאה:

1. #define N 100*1024*1024;

```

2.  int a[N];
3.
4.  int main() {
5.      if (fork() > 0) { // parent
6.          wait(NULL);
7.      } else { // child
8.          printf("Bye!\n");
9.      }
10.     return 0;
11. }

```

2. כמה חריגות דף (page faults) יוצר תהליך הבן מתחילתו ועד סופו (שורות 7-10)?
הניחו כי כל קריאות המערכת מצליחות.

- (a) 0, בגלל שמנגנון COW חוסך את כל העתקות הזיכרון מהאב לבן.
- (b) לפחות 1, בגלל ניסיון כתיבה של הבן למחסנית המשתמש שלו
- (c) לפחות 1, בגלל ניסיון כתיבה של הבן למחסנית הגרעין שלו
- (d) לפחות 1, בגלל ניסיון כתיבה של הבן לערימה שלו
- (e) אולי 0 ואולי יותר, תלוי במספר ההחטאות ב – TLB
- (f) אולי 0 ואולי יותר, תלוי בתזמון היחסי בין תהליך האב לתהליך הבן.

נמקו:

יובל, שהיה קצת מבולבל מכל החומר שהוא ספג במהלך הקורס, לא זכר מדוע משתמשים במנגנון COW. לכן

הוא שינה את קוד הגרעין של לינוקס וביטל את מנגנון COW, כך שקריאת המערכת fork מעתיקה מיד את כל איזורי הזיכרון הנגישים לכתיבה מתהליך האב לתהליך הבן. לאחר מכן יובל השווה בין:

- $T(COW) =$ זמן הריצה של התכנית הנ"ל במערכת ההפעלה המקורית, עם מנגנון COW.
- $T(NO-COW) =$ זמן הריצה של התכנית הנ"ל במערכת של יובל, ללא מנגנון COW.

כדי להוריד את רעש המדידה, יובל הקצה שתי ליבות מעבד באופן אקסלוסיבי לשני התהליכים, האב והבן, כך שהם ירוצו ללא עיכובים מיותרים.

3. באיזה מערכת זמן הריצה יהיה קצר יותר, ומדוע?

- a. $T(COW) > T(NO-COW)$ בגלל שמעתיקים פחות זיכרון מהאב לבן.
- b. $T(COW) < T(NO-COW)$ בגלל שמעתיקים פחות זיכרון מהאב לבן.
- c. $T(COW) > T(NO-COW)$ בגלל שמספר חריגות הדף קטן יותר.
- d. $T(COW) < T(NO-COW)$ בגלל שמספר חריגות הדף קטן יותר.
- e. $T(COW) > T(NO-COW)$ בגלל שתהליך האב יכול לרוץ במקביל לתהליך הבן.
- f. $T(COW) < T(NO-COW)$ בגלל שתהליך האב יכול לרוץ במקביל לתהליך הבן.

נמקו:

```

1. #define N 100*1024*1024;
2. int a[N];
3.
4. int main() {
5.     if (fork() > 0) { // parent
6.         wait(NULL);
7.     } else { // child
8.         for (unsigned int i = 0; i < N; ++i) {
9.             a[i] = 1;
10.        }
11.    }
12.    return 0;
13. }

```

4. כמה חריגות דף יוצר תהליך הֶבֶן מתחילתו ועד סופו (שורות 7-12)? בחרו את התשובה הקרובה ביותר מבין הבאות (אנחנו מתעניינים בסדר גודל ולא תשובה מדויקת).
הניחו כי כל קריאות המערכת מצליחות.

- 1024*1 (a)
- 1024*10 (b)
- 1024*100 (c)
- 1024*1024 (d)
- 1024*1024*10 (e)
- 1024*1024*100 (f)

נמקו:

5. יובל חזר על תהליך המדידה שתואר קודם גם עבור התכנית החדשה. באיזה מערכת זמן הריצה יהיה קצר יותר, ומדוע?

- a. $T(COW) > T(NO-COW)$ בגלל שמעתיקים פחות זיכרון מהאב לבן.
- b. $T(COW) < T(NO-COW)$ בגלל שמעתיקים פחות זיכרון מהאב לבן.
- c. $T(COW) > T(NO-COW)$ בגלל שמספר חריגות הדף קטן יותר.
- d. $T(COW) < T(NO-COW)$ בגלל שמספר חריגות הדף קטן יותר.
- e. $T(COW) > T(NO-COW)$ בגלל שתהליך האב יכול לרוץ במקביל לתהליך הבן.
- f. $T(COW) < T(NO-COW)$ בגלל שתהליך האב יכול לרוץ במקביל לתהליך הבן.

נמקו:

שאלה 3

שאלה זו כתובה באנגלית מאחר ושהיא מכילה לא מעט מושגים ושמות אשר קל יותר לבטאם באנגלית. פתרו סעיף זה באיזה שפה שתמצאו.

In the wet part of this homework, you implemented an interface that manages dynamic memory for a process.

In this part of the homework, you will analyze the existing `malloc()` (from `<cstdlib>`) while learning about some new Linux tools.

NOTE: Do NOT submit code you write in this homework with your wet submission. Simply copy your code to your dry submission file, wherever requested.

1. Look up the "strace" utility online, read a little bit, and try to use it yourself by running ``strace ls`` in your OS terminal. Finally, explain here in a few words what it does.

-
-
-
2. Write a simple program in C that receives a number “x” from the command line and allocates (using malloc()) a block of memory that is “x” bytes long. You can assume there’s always one input it will always be a positive integer. Run strace with your compiled program.

Finally, **attach the code of the program** and a **screenshot** of the output of running strace with your compiled program below.

3. The output you received from running strace on your program was probably very messy. There’s no way to tell which system call was used during the execution of malloc. Suggest a simple addition to your C code, such that you will be able to spot the system call used during the execution of malloc anyway. You’re not allowed to add flags to strace. Your change must be made in the C code.

Finally, **attach the code of the updated program here**:

4. In the wet part of this homework, you wrote/will write a malloc() alternative that uses both sbrk() and mmap().

Your job in this section is to determine which memory related functions/system calls the **original** malloc() function that is included in stdlib uses.

Hint: Use the program and the tools from the last sections to help you out!

Which two system calls does the stdlib standard malloc() use in its implementation? Attach **screenshots** that prove your answer.

5. **Find the threshold** that malloc uses to transition from using one function to the other. In other words, what is the number of bytes, after which calling malloc with that number, would result in using one system call instead of the other? Attach **screenshots** that prove your answer.
-
