

Operating Systems – 234123 **Homework Exercise 4 – Dry**

Teaching Assistant in charge:

# Bahjat Kawar

Assignment Subjects & Relevant Course material **Virtual Memory & Memory Management Recitations 9-10, Lecture 8**

Page 1 of 8

## 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 –

**DHW2\_123456789\_300200100.pdf**

1. The submission should contain the following:
   1. The first page should contain the details about the submitters - Name, ID number and email address.
   2. Your answers to the dry part questions.
2. Submission is done electronically via the course website, in the **HW2 – 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. Convoluted 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/4qJZK6K6zLQa7uS79>

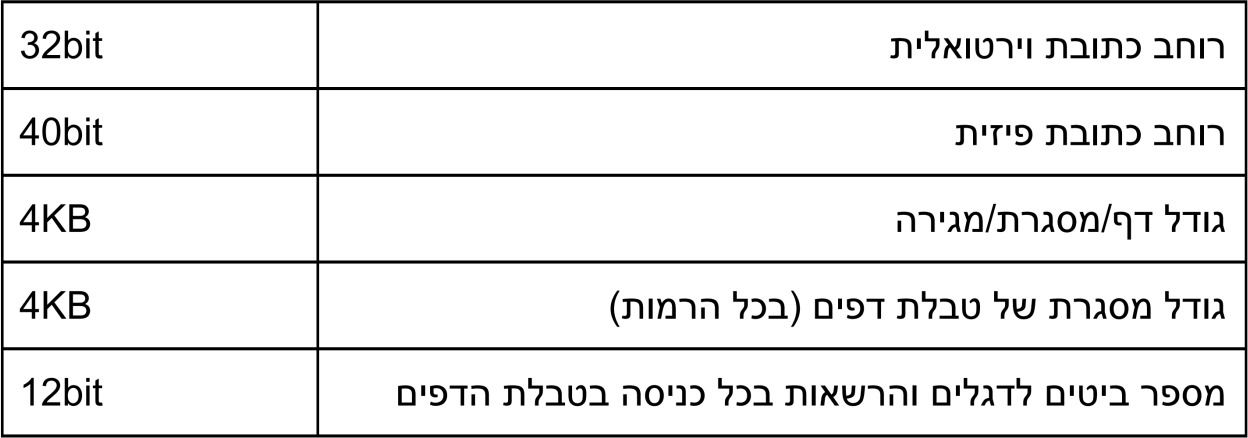
Page 2 of 8

# שאלה 1 - זיכרון:

עדן, זמרת פופולרית, סבלה ממחסור בזיכרון פיזי במחשב שלה (בעל מעבד IA-23 וזיכרון פיזי בגודל )GB4 ולכן הציעה תכן חדש של מעבד המרחיב את מרחב הזיכרון הפיזי מ23- ל44- ביט. כתוצאה מכך, במימוש של

עדן יש שלוש רמות תרגום בטבלת הדפים.

שאר נתוני המעבד של עדן זהים לאלו של מעבד IA-23 ,כלומר נתוני המערכת החדשה הם:



.1 בהנחה שגודל כניסה בטבלת הדפים **מעוגל למעלה לחזקה שלמה של** ,**2** מהו אופן חלוקת הכתובת

הוירטואלית לשדות בתהליך תרגום כתובות walk) (page ?



נימוק:

נשים לב כי גודל המסגרת הפיזית הינו kb4 ועל כן עלינו לדרוש offset שיוכל לגשת לכל המסגרת. לכן, נדרוש כי בכתובת הוירטואלית יהיו b12 של offset אשר יועברו לכתובת הפיזית. גודל הכתובת הפיזית הוא b40 ומכאן נסיק כי בהינתן b12 של offset, נותרו 40b-12b=28b שייצגו את הכתובת בה מתחילה המסגרת הפיזית. לכן, בכל כניסה בטבלת הדפים, עלינו לשמור b12 של דגלים (כמוגדר לעיל) ו-b28 לטובת ייצוג הכתובת הפיזית של תחילת המסגרת ומכאן שגודל כניסה כזו הינו **לפחות** 12b+28b=40b. מהנתון כי גודל כניסה בטבלת הדפים מעוגל למעלה לחזקה שלמה של 2, נסיק כי גודל כניסה הינו **64b=8byte**. בנוסף, נתון כי גודל דף היו 4kb וראינו כי גודל כל כניסה 8byte ולכן ישנן סה"כ 4kb/8byte=512 כניסות ולכן נדרוש כי בכתובת הוירטואלית יהיו ל-index1 ו-index2 9b לטובת ייצוג כניסות אלה (512=2^9). השתמשנו ב-12b+9b+9b=30b מתוך הכתובת הוירטואלית שרוחבה 32b ועל כן נשתמש ב-2b האחרונים לייצוג הרמה העליונה וכך נשלים ל-32b. **(התשובה היא א')**.

Page 3 of 8

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 .3

.3 מהו הגודל המינימלי האפשרי של כניסה בטבלת הדפים אם **לא מעגלים אותו למעלה**?

* 1. 2 בתים
  2. 4 בתים
  3. 5 בתים
  4. 6 בתים
  5. 7 בתים
  6. אף תשובה אינה נכונה

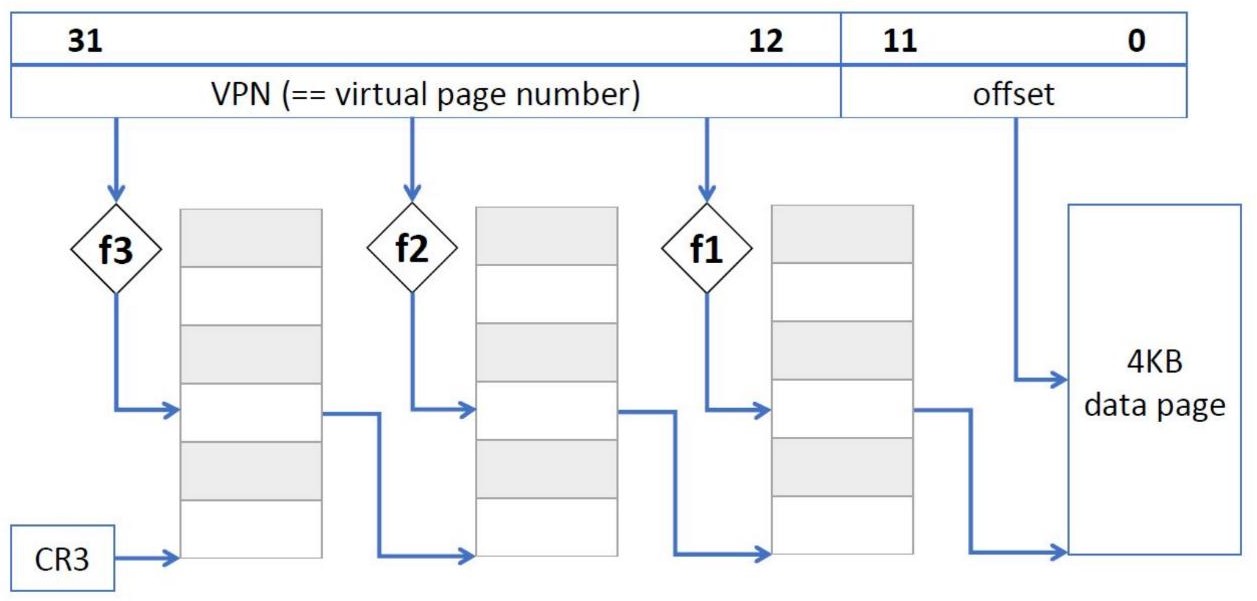
נימוק:

כפי שראינו בסעיף קודם, גודל כניסה מינימלי הנדרש לנו לשמירת המידע הנחוץ הינו 40b=5byte.

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 f3( f2, )f1, המקבלות את מספר הדף הוירטואלי VPN ומחזירות, בהתאמה, שלושה אינדקסים לשלושת הרמות בטבלת הדפים. בכל הסעיפים הבאים, הפעולות חלוקה

/ ומודולו % הן פעולות בשלמים. למשל:

1024/819=1

1024%819=205

Page 4 of 8

.2 מהי הפונקציה ?f1

f1(vpn) = vpn/819 .a f1(vpn) = vpn%819 .b f1(vpn) = (vpn/819)/819 .c f1(vpn) = (vpn%819)/819 .d f1(vpn) = (vpn/819)%819 .e

f1(vpn) = ((vpn/819)%819)/819 .f

נימוק:

כאשר מגיעים לרמה הנמוכה ביותר, עלינו להחזיר את הכתובת לתחילת המסגרת הפיזית בה נמצא המידע שאנו צריכים. נשים לב כי ישנן 4kb/5byte=819 כניסות בכל רמה ובפרט ברמה התחתונה ביותר. ישנן 819 מסגרות ולכן עלינו להיות יכולים לגשת לכל המסגרות בטווח 0-818 ונשיג זאת בעזרת f(vpn)=vpn%819 אשר נותנת תשובות רק בטווח זה.

.4 מהי הפונקציה ?f2

f1(vpn) = vpn/819 .a f1(vpn) = vpn%819 .b f1(vpn) = (vpn/819)/819 .c f1(vpn) = (vpn%819)/819 .d f1(vpn) = (vpn/819)%819 .e

f1(vpn) = ((vpn/819)%819)/819 .f

נימוק:

כפי שראינו בהרצאות ובתרגולים, פעולת חלוקה על מספר זהה להזזב ימינה של הביטים בייצוג הבינארי של מספר זה. נרצה "להיפטר" מהביטים הראשונים שייצגו את הכניסות של f1 ולאחר מכן נמצא את שארית החלוקה ב-819 כדי לקבל index כניסה בטווח 0-818 כפי שראינו בסעיף קודם.

Page 5 of 8

.5 מהי הפונקציה ?f3

f1(vpn) = vpn/819 .a f1(vpn) = vpn%819 .b f1(vpn) = (vpn/819)/819 .c f1(vpn) = (vpn%819)/819 .d f1(vpn) = (vpn/819)%819 .e

f1(vpn) = ((vpn/819)%819)/819 .f

נימוק:

ראינו כי החלוקה הראשונה "תיפטר" מהביטים שייצגו את הכניסות ברמה הנמוכה ביותר. כעת נבצע חלוקה נוספת על מנת "להיפטר" מהביטים המייצגים את הכניסות ברמה השנייה. נשים לב כי 1G<(819^3) ועל כן החלוקה השניה תיתן לנו בהכרח מספר בטווח 0-818, למעשה נקבל מספר בטווח 0-4 וזה עומד בדרישות אשר הצבנו בסעיף א' לרמה זו.

.6 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 הTLB- 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:

במערכת אותה הציע שוקי נדרשים פחות בתים כדי לייצג את המידע בכל כניסה בדף. על כן, בדף בגודל 4kb נוכל לשמור 819 כניסות במקום 512 כניסות כפי שראינו בפתרון של עדן. מכאן ניתן להסיק כי נוכל לכסות יותר זיכרון ב-TLB בעזרת הפתרון של שוקי מאשר בעזרת הפתרון של עדן.

Page 6 of 8

# שאלה 2 – ניהול זיכרון:

שאלה זו כתובה באנגלית מאחר ושהיא מכילה לא מעט מושגים ושמות אשר קל יותר לבטאם באנגלית. פתרו

סעיף זה באיזה שפה שתרצו.

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

### Section 1:

* 1. Look up the “strace” utility online, read a little bit, and try to use it yourself by running

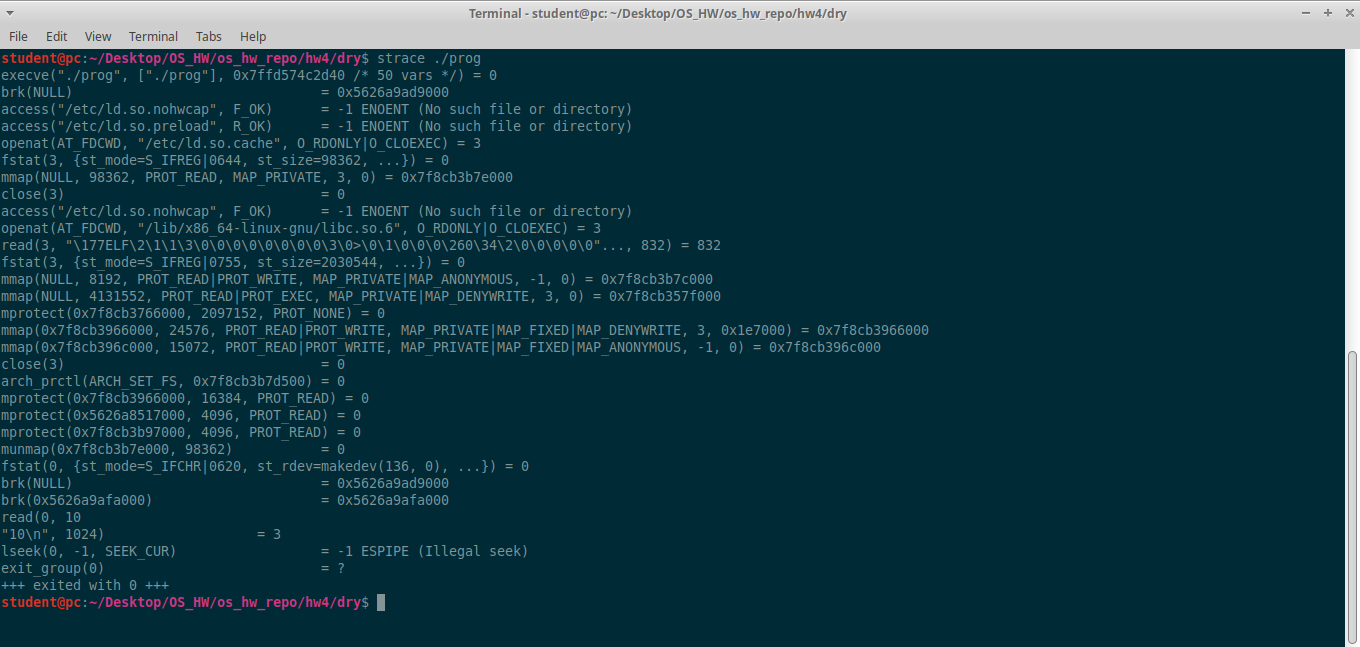
`strace ls` in your OS terminal. Finally, explain here in a few words what the it does.

strace runs the specified command until it exits. It records the system calls which are called by a process and the signals which are received by a process. The name of each system call, its arguments and its return value are printed on standard error or other specified file descriptor.

* 1. 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

Output:



Code:

#include<stdio.h>

#include<stdlib.h>

int main() {

size\_t bytes = 0;

/// get 'x' from user. (we assume we get 1 positive number)

scanf("%lu", &bytes);

/// allocate block of size 'x'

void\* ptr = malloc(bytes);

return 0;

}

Page 7 of 8

* 1. 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.

### Section 2:

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 functions the malloc() function that is included in your stdlib uses.

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

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

1) 2)

1. 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.

Page 8 of 8