Operating Systems – 234123

Homework Exercise 3 – Dry

Submitters:

Tom Guy

315155671

[tom.guy@campus.technion.ac.il](mailto:tom.guy@campus.technion.ac.il)

Nikita Ivanov

322245101

[Nikitaiv@campus.technion.ac.il](mailto:Nikitaiv@campus.technion.ac.il)

**שאלה 1 - זיכרון :**

עדן, זמרת פופולרית, סבלה ממחסור בזיכרון פיזי במחשב שלה (בעל מעבד 32-IA וזיכרון פיזי בגודל (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 32-,IA 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 |

## סעיף 1

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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:

**תשובה:**

ראשית, מכך שגודל הדף נשאר זהה – 4KB, גם החישוב של מציאת offset בתוך הדף יישאר זהה ולכן דרושים 12 ביטים למציאת מיקום הכתובת הפיזית המבוקשת בתוך הדף. כעת נבין את ההיררכיה הרצויה על מנת לתמוך בכתובות פיזיות בגודל 40 ביטים. גודל כל כתובת מעוגל לחזקה שלמה של 2 ולכן יהיה 64 ביטים. גודל כל page table הוא כגודל דף – 4KB ומכך שכל כתובת בו היא 64 ביטים = B8 יש PTE’s ולכן נצטרך 9 ביטים כדי למפות כל PTE ב-page table. יש לנו מספיק ביטים כדי לעשות 2 כאלה ול- האחרון יישארו 2 ביטים.

סעיף 2

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.

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

a) 3 בתים b) 4 בתים c) 5 בתים d) 6 בתים e) 7 בתים

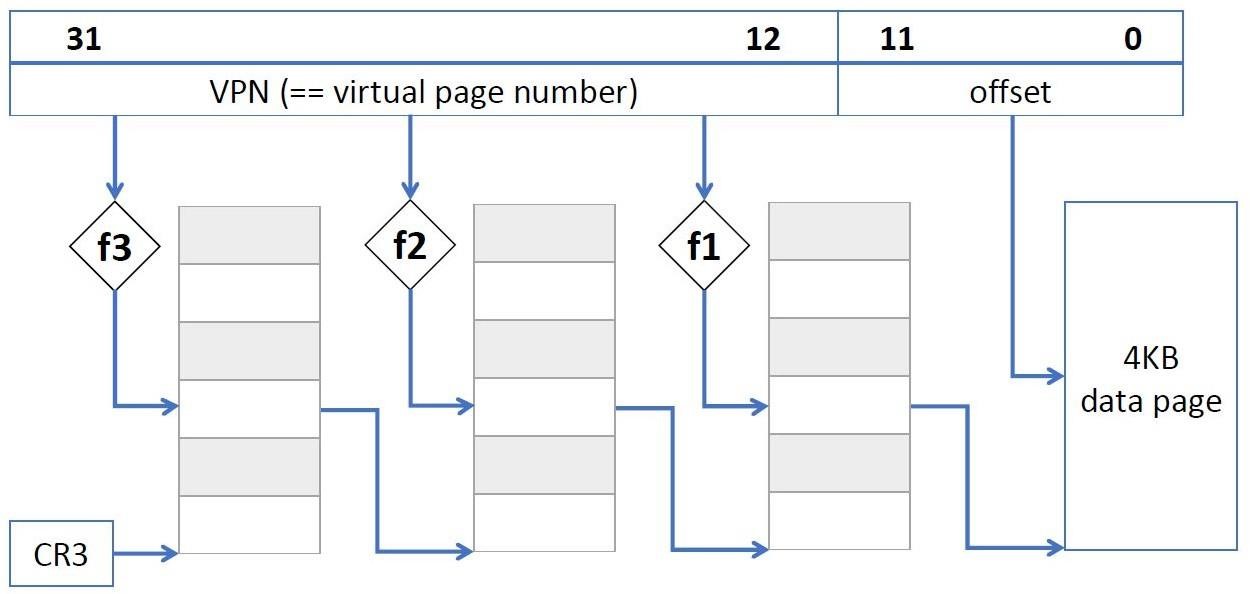
f) אף תשובה לא נכונה נימוק:

5B = 40b ולכן כל כניסה בטבלת הדפים שמכילה 12 ביטים של דגלים ו- 28 (=40-12, 12 ביטים של offset), ביטים למיפוי כתובת יכולה להישמר בלפחות חמישה בייטים.

## סעיף 3

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 הווירטואלי נשמרים בכניסות סמוכות בטבלת הדפים. שוקי הבחין שבמימוש החדש הכתובת הווירטואלית 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 הווירטואלי 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

מהי הפונקציה ?f1 **נמקי**.

**תשובה: לסעיף הנוכחי ולשניים אחריו**

מסעיף קודם, ראינו כי גודלה של כל כתובת (כניסה בטבלה) הוא 5 בייטים ולכן יש סה"כ כתובות אפשריות בכל דף – כמובן שנשתמש במספר שלם של כתובות ולכן יש סה"כ 819 כתובות בדף. את המיפוי עבור כתובת וירטואלית כללית נבצע בצורה הבאה:

* נציג את הכתובת בצורה הבאה: כאשר
* המיפוי לפונקציות יהיה:

מסעיף קודם, ראינו כי כל כתוב

סעיף 4 מהי הפונקציה ?f2 **נמקי**.



**תשובה:**

ענינו גם על סעיף זה מעלה.

סעיף 5מהי הפונקציה ?f3 **נמקי.**



**תשובה:**

ענינו גם על סעיף זה מעלה.

סעיף 6

מה היתרון של המערכת שהציע שוקי על פני המערכת שהציעה עדן? **נמקי.**

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:

במימוש של עדן, כל טבלת דפים הכילה 512 PTEs’ לכל היותר (עבור ) בעוד במימוש של שוקי כל טבלה מכילה 819 PTEs’.

## 

## סעיף 7

לכל אחד מהסעיפים הבאים, סמני האם התשובה נכונה או לא, **ונמקי**.

הגדרה: מצביע **חוקי** הינו כזה שמצביע על כתובת בזיכרון שנמצאת בתוך אזור המוקצה בשביל התהליך\חוט גרעין\..

שמחזיק את המצביע.

1) אם למחשב יש 16GB של **RAM**, יתכן שלתוכנית **משתמש** כלשהי יש מצביע חוקי לכתובת הגדולה מ-

.)HEX (בסיסOx200000001

רמז: האם הכתובת ששמורה בתוך מצביע היא פיזית או ווירטואלית ?

תשובה: נכון \ לא נכון.

נימוק:

הכתובת היא מצביע בתוכנית - כתובת וירטואלית ולכן יתכן בהחלט כי המשתמש יקבל אותה בתור מצביע בעוד שהיא לא קיימת בזיכרון הפיזי, אך היא תמופה לכתובת פיזית הקטנה מ-.

2) אם למחשב יש 16GB של זיכרון **ווירטואלי**, יתכן שיש מצביע חוקי (לא בהכרח לתהליך משתמש) לכתובת הגדולה מ 16,000,000,000 (בסיס דצימלי).

תשובה: נכון \ לא נכון.

נימוק:

ייתכן בהחלט. לדוגמה, אם גודל כתובת פיזית גדול מגודל כתובת וירטואלית (יש יותר זיכרון פיזי מוירטואלי), כמו לדוגמה בסעיף א' (רק עם גודל כתובת שונה) יתכן כי הכתובות הגדולות מ-16,000,000,000 קיימות ותהליכי גרעין שנגישים לכתובות בצורה ישירה ללא זיכרון וירטואלי ייגשו אליהן בצורה חוקית כחלק מריצת תהליך.

3) בהינתן מחשב גודל דף = גודל מסגרת = 2KB, יתכן שהכתובת הוויראלית Ox800 ו Ox808 ממפות לכתובת במרחק גדול מ Ox8 אחת מהשנייה בזיכרון הפיזי.

תשובה: נכון \ לא נכון.

נימוק:

נשים לב כי כעת גודל הדף הוא ולכן גודל ה- חייב להיות ביטים.   
עבור ועבור כל הביטים אחרי הביט ה-11 שווים ורק ה- ביניהם שונה. לכן הם ממופים לאותו **דף**. בגלל שההפרש ב- הוא 8, ההפרש בין שתי הכתובות הפיזיות הוא בדיוק 8.

4) מנגנון הזיכרון הווירטואלי (כמו שהוא) מאפשר לנו לדמות **דיסק (**אחסון עמיד) גדול יותר ממה שיש לנו בפועל.

תשובה: נכון \ לא נכון.

נימוק:  
נשים לב כי אמנם חלק מהמטרה של זיכרון וירטואלי הוא לדמות שמירת זיכרון רב יותר ממה שיש למחשב לתת, אבל זה לא אפשרי לאחסן יותר מידע מכמות הזיכרון הפיזי שיש לנו ולכן לא יווצר מצב ששמור לנו יותר מידע בזכרון הווירטואלי מאשר כמות הזיכרון הפיזי (ושמירת חלק מהמידע על ה- לא נחשב כדימוי לדיסק גדול יותר).

**שאלה 2 – ניהול זיכרון :**

שאלה זו כתובה באנגלית מאחר ושהיא מכילה לא מעט מושגים ושמות אשר קל יותר לבטאם באנגלית. נא לפתור אותה באיזה שפה שתרצנה\ו.

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 it does.   
     
   The “strace” utility collects all the system calls the specified command used while running, and returns them out after the command exists.

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   


**Text

Description automatically generated**

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.

ננצל את העובדה ש-strace מנסה להדפיס את פקודות המערכת שהיא מקבלת ברצף ונכניס שתי פקודות write לפני ואחרי ביצוע ה-malloc ובכך נוכל למצוא בקלות את קריאות המערכת בהם השתמשה התוכנית בעת ביצועו.  
****

**Text

Description automatically generated  
  
  
  
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.

Text

Description automatically generated 1) brk 2) mmap

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.

**תשובה: 134537**

התשובה הנתונה פה מתייחסת רק למקרה בו ה-malloc שעשינו הוא הראשון בתוכנית וכמות הבייטים היא הראשונה בה נשתמש ב-mmap בנוסף ל-brk. במקרה זה, מכיוון שמערכת ההפעלה דואגת ליישור הכתובת, היא תקצה יותר מקום מהדרוש – כמה שביקשנו ועוד כמות הזיכרון שמערכת ההפעלה מקצה לשימושה בכל בלוק – כמו מה שאנחנו מימשנו בתרגיל. באופן כללי, ה-threshold דינמי ומשתנה במהלך ריצת התוכנית, אך מערכת ההפעלה לא תקצה פחות מ-135168 בייטים בעת שימוש ב-mmap. כל זאת למרות שב-mmap man כתוב כי ה-threshold מוגדר להיות 131072=128KB דיפולטיבית.

Text

Description automatically generated

Text

Description automatically generated