

Operating Systems – 234123

Homework Exercise 4 – Dry

Teaching Assistant in charge:

**Or Keret**

Assignment Subjects & Relevant Course material **Virtual Memory & Memory Management Recitations 10-12, Lectures 8-9**

Page 1 of 6

## 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

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 **HW4 – Dry** submission box.

## Grading

1. **All** question answers must be supplied with a **full explanation**. Most of your grade is determined by your **explanation** and **evident effort**, and not on the absolute correctness of your answer.
2. Remember! your goal is to communicate the knowledge that you have acquired. Full credit will be given only to correct solutions which are **clearly** described. Convoluted and obtuse descriptions will receive low grades.

## 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.office.com/r/5mwQE7EGzm>

Page 2 of 6

# שאלה 1 – ניהול זיכרון:

נתון שהמחשב זה עתה עלה (מיד אחרי ,)reboot שהארכיטקטורה של המחשב היא ,x86/64bit שמשתמשת יחידה

בשם אליס משתמשת בו כרגע, שכל מה שאליס עשתה עד עתה זה להריץ shell על המחשב (כחלק מתהליך ה- ,)login שקיים קובץ בשם my\_file.txt בתיקיית העבודה הנוכחית של אליס בגודל ,32KB ושהתהליך הראשון

שאליס מריצה בshell- מבצע את קטע הקוד הבא:

|  |  |
| --- | --- |
| 1 | #define PAGE\_SIZE (4\*1024) |
| 2 | int main()  { |
| 3 | int fd = open("./my\_file.txt", O\_RDWR); |
| 4 | char\* buffer = malloc(2 \* PAGE\_SIZE); |
| 5 | read(fd, buffer, 2 \* PAGE\_SIZE); |
| 6 | char\* array = (char\*) mmap(NULL, 4 \* PAGE\_SIZE, PROT\_READ | PROT\_WRITE, MAP\_SHARED, fd, 0); |
| 7 | char x = array[12]; |
| 8 | char z = array[12 + 2 \* PAGE\_SIZE]; |
| 9 | array[12 + 2 \* PAGE\_SIZE] = x; |
| 10 | return 0; |
| 11 | } |

.1 מה המספר **המינימלי** של מסגרות פיזיות חדשות שמוקצות (בעבור data בלבד, מבלי להתחשב בtable- )page

בכל אחת מהשורות הבאות?

|  |  |  |
| --- | --- | --- |
| הסבר קצר | מספר מסגרות | שורה |
| Open לא מקצה מסגרות פיזיות. | 0 | 3 |
| הקצאת המסגרות תתבצע רק בעת גישה ל-buffer.(מנגנון עצלני) | 0 | 4 |

Page 3 of 6

|  |  |  |
| --- | --- | --- |
| יוקצו 2 מסגרות עבור ה-buffer ו-2 מסגרות עבור המידע שקוראים מהקובץ אל מטמון הדפים. | 4 | 5 |
| תתבצע הקצאה רק לאחר page fault שיתקבל בעת גישה למידע. | 0 | 6 |
| המסגרת הראשונה כבר קיימת בזיכרון במטמון הדפים משורה 5. | 0 | 7 |
| גישה ראשונה לדף השלישי של array יוצרת page fault שמקצה מסגרת חדשה. | 1 | 8 |
| גישה שנייה לדף השלישי כבר לא מקצה מסגרת חדשה מאחר והמיפוי של הקובץ הוא משותף. | 0 | 9 |

.2 כיצד תשתנה תשובתכן לסעיף 1 אם בשורה 6 היה רשום MAP\_PRIVATE במקום ?MAP\_SHARED ציינו

באיזה שורות תשובתכן הייתה **משתנה** והסבירו:

### תשובה:

### התשובה תשתנה עבור שורה 9 כי כעת אנו מנסים לכתוב לדף השלישי ומכיוון שהאזור הוגדר להיות פרטי אז בכתיבה הראשונה (Copy-On-Write)נקבל page fault ממערכת ההפעלה שתקצה לנו מסגרת אנונימית חדשה ותעתיק אליה את המידע מהמסגרת הפרטית.

.3 כיצד תשתנה תשובתכן לסעיף 1 אם מיד כאשר התהליך הנ”ל מסתיים הוא מורץ שוב? נמקו (השאלה מתייחסת

להרצה השנייה).

### תשובה:

### בשורה 5 יוקצו רק 2 מסגרות עבור ה-buffer כי המסגרות של הקובץ שהוקצו בהרצה הראשונה יהיו עדיין קיימות ולא יהיה צורך להקצות חדשות.

.4 כאשר מורץ הקוד לראשונה, מה הוא המספר **המקסימלי** של מסגרות פיזיות חדשות שמוקצות בעבור **טבלת**

### הדפים של התהליך? יש לשרטט את המסגרות של טבלאות הדפים בשביל נימוק התשובה.

הנחות: )1( התעלמו ממסגרות המוקצות עבור המחסנית .)stack(

)2( הניחו שהכתובת של buffer (שורה )4 היא:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 47 39 | 38 30 | 29 21 | 20 12 | 11 0 |
| 000000000 | 000000**1**00 | 111111111 | 111111111 | 000000000000 |

)3( הניחו שהכתובת של array (שורה )6 היא: (שימו לב להבדל בכתובות)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 47 39 | 38 30 | 29 21 | 20 12 | 11 0 |
| 000000000 | 000000**0**00 | 111111111 | 111111111 | 000000000000 |

### תשובה:

.major page fault הגדירו .5

### תשובה: page fault שדורש גישה לדיסק בכדי לטפל בו (התהליך יוצא להמתנה), קורה במקרה של מיפוי מגובה קובץ.

.6 מה מספרי השורות בהן מתרחש fault page ?major נמקו

### תשובה:

### בשורה 5 מתרחש major page fault כי אנחנו צריכים לגשת לקובץ כדי לקרוא ממנו ולהכניס שני דפים למטמון הדפים.

### בשורה 8 – צריך להביא את הדף השלישי למטמון הדפים ולכן צריך גישה לקובץ.

.7 כיצד תשתנה תשובתכן לסעיף **6** אם בשורה 6 היה רשום MAP\_PRIVATE במקום ?MAP\_SHARED נמקו

### תשובה:

תשובתנו לא תשתנה, מכיוון שכתיבה לדיסק בשורה 9 לא הייתה דורשת גישה לדיסק אלא רק למטמון הדפים ושאר הקריאות לא ישנו את השורות בהן מתרחש page fault.

# 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 the it does.

Strace prints out the signals and system calls that where called while running the program that is passed to it, it also prints the arguments and return values of the system calls, it is useful to use as a debugging toll.

* 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

A computer screen shot of a program

Description automatically generated with low confidence

Page 5 of 6

* 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.

One change we can do is to “wrap” the malloc call with two other system calls, sleep for example.  
We can call sleep(1) just before the malloc call and then once again sleep(1) just after the malloc call.  
That was when we compile with strace we can see exactly where the sleep starts (both because the command line “stutters” for one second and because we can see in the final strace when was sleep called and infer from that what were the exact system calls that where used by our malloc.

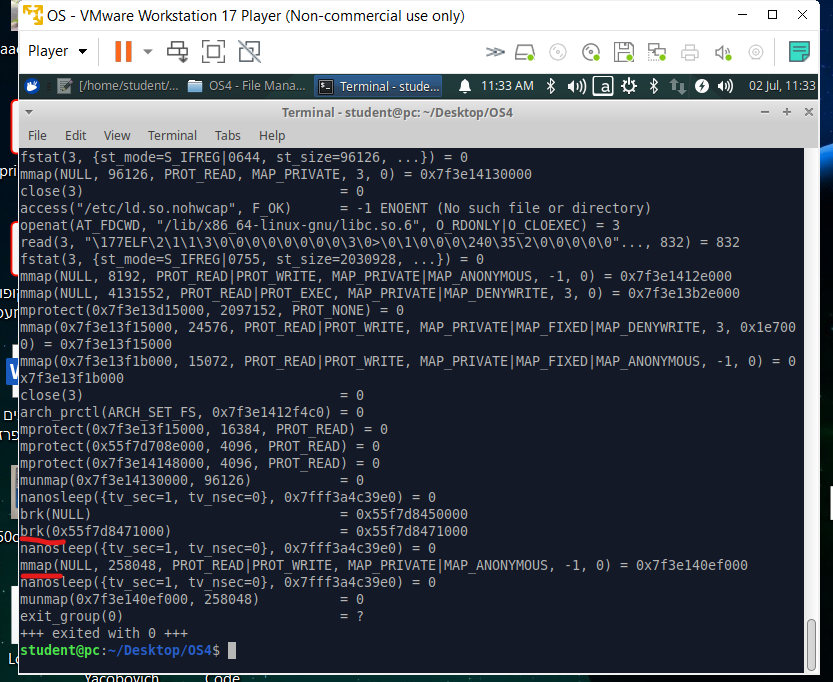
### Section 2:

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!

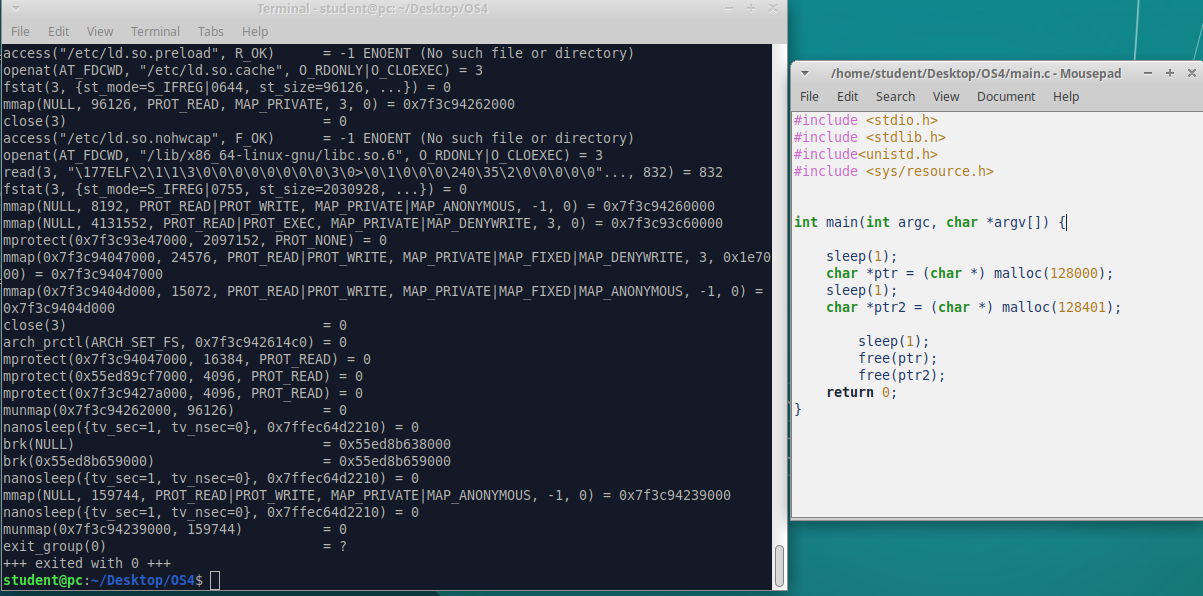
1. Which **two** system calls does the stdlib standard malloc() use in its implementation? Attach screenshots that prove your answer.

1)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.

128KB (as seen in the manual).





Page 6 of 6