

Operating Systems – 234123

Homework Exercise 4 – Dry

Presented by:

Nadav Orzech 311549455

Roni Englander 312168354

Emails:

nadav.or@campus.technion.ac.il

roni.en@campus.technion.ac.il

שאלה 1 - זיכרון:

עדן, זמרת פופולרית, סבלה ממחסור בזיכרון פיזי במחשב שלה (בעל מעבד IA-32 וזיכרון פיזי בגודל GB4) 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. בהנחה שגודל כניסה בטבלת הדפים **מעוגל למעלה לחזקה שלמה של 2**, מהו אופן חלוקת הכתובות הוירטואליות 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: (תשובה א)

מכיוון שנתון שגודל דף נותר 4kb, נצטרך 12 ביטים למפות את הדף, ולכן ה-offset הוא 12. השינוי שעדן עשתה הוא לשנות את רוחב הכתובת הפיזית מ-32 ביטים ל-40 ביטים. מכיוון שנתון שגודל כניסה בטבלת הדפים מעוגל למעלה לחזקה שלמה של 2, נדרשים 64 ביטים לייצג כתובת פיזית, כאשר 24 ביטים החזקים אינם בשימוש. (64 ביטים=8 בתיים)

נתון שגודל מסגרת של טבלת הדפים הוא 4kb, ולכן בטבלה יש $4kb/8b=512$ כניסות. לכן על מנת למפות כל טבלה יש צורך ב-9 ביטים ולכן כל אינדקס יהיה בגודל 9. ייצוג של מילה הוא 32 ביטים ולכן בשביל שתי הרמות התחתונות נשתמש ב-9 ביטים, ונותר לרמה העליונה 2 ביטים.

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.

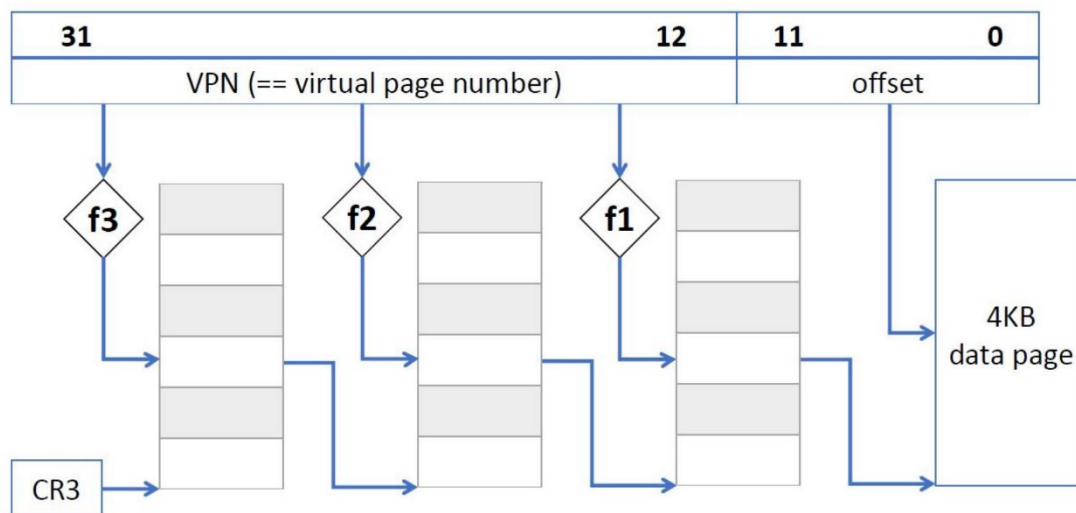
2. מהו הגודל המינימלי האפשרי של כניסה בטבלת הדפים אם לא מעגלים אותו למעלה?

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

נימוק: (תשובה c)

נתון שרוחב כתובת פיזית הוא 40 ביטים (כאשר 12 התחתונים משמשים ל frame offset ו-28 העליונים משמשים לframe number), כלומר 5 בתים. לכן הגודל המינימלי האפשרי של כניסה בטבלת הדפים הוא גודל הייצוג של כתובת פיזית – 5 בתים.

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:

$$1024/819=1$$

$$1024\%819=205$$

3. מהי הפונקציה f1?

a. $f1(vpn) = vpn/819$

b. $f1(vpn) = vpn\%819$

- c. $f1(vpn) = (vpn/819)/819$
- d. $f1(vpn) = (vpn\%819)/819$
- e. $f1(vpn) = (vpn/819)\%819$
- f. $f1(vpn) = ((vpn/819)\%819)/819$

נימוק: (תשובה b)

במעבד הנתון יש 4kb/5b כניסות בטבלת הדפים, כלומר 819 כניסות בכל רמת תרגום (בדומה ל-1024 במעבד הרגיל). לכן נרצה עבור $f1$ לשלוח את שארית החלוקה של ה-VPN ב-819, מספר שנמצא בתחום בין 0 ל-818, על מנת לקבל מספר כניסה בטבלה הראשונה שתחזיק את מספר המסגרת הרלוונטית. נבחים כי החלוקה שנבצע לא תחלק ביטים שונים עבור כל רמת תרגום, ויהיו ביטים שנשתמש למספר תרגומים.

4. מהי הפונקציה $f2$?

- a. $f1(vpn) = vpn/819$
- b. $f1(vpn) = vpn\%819$
- c. $f1(vpn) = (vpn/819)/819$
- d. $f1(vpn) = (vpn\%819)/819$
- e. $f1(vpn) = (vpn/819)\%819$
- f. $f1(vpn) = ((vpn/819)\%819)/819$

נימוק: (תשובה e)

באותו אופן כמו בסעיף הקודם ובדומה למימוש במעבד הרגיל, נבצע ראשית חלוקה ב-819 ולאחר מכן נחשב שוב את שארית החלוקה לתוצאה. מכיוון שכפי שצינו, החלוקה לא מחלקת את הביטים של ה-VPN לרמות שונות בצורה מוחלטת, ולכן לא מספיק לבצע את פעולת החלוקה בלבד. לכן החלוקה ב-819 לא בהכרח תחזיר ערך בין 0 ל-818 אז נבקש שוב להוציא את השארית.

5. מהי הפונקציה $f3$?

- a. $f1(vpn) = vpn/819$
- b. $f1(vpn) = vpn\%819$
- c. $f1(vpn) = (vpn/819)/819$
- d. $f1(vpn) = (vpn\%819)/819$
- e. $f1(vpn) = (vpn/819)\%819$
- f. $f1(vpn) = ((vpn/819)\%819)/819$

נימוק: (תשובה c)

בהמשך לדברים שנאמרו לעיל, נחלץ את הרמה האחרונה על ידי חלוקה ב-819 פעמיים. כעת מובטח לנו כי הערך שנקבל יהיה בתחום שאנחנו רוצים (מכיוון שגודלו של ה-VPN הוא 2^{20}) ויוכל לשמש אותנו לתרגום הרמה העליונה ביותר.

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. 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: (תשובה d)

במימוש של שוקי כל טבלת דפים מייצגת 819 כניסות, ואילו במימוש של עדן כל טבלת דפים מייצגת 512 כניסות (כפי שחישבנו בסעיפים הקודמים). המשמעות היא שכל טבלה אצל שוקי מייצגת מרחב זיכרון רחב יותר ולכן נדרשות פחות מסגרות במימוש של שוקי, דבר הגורם לכך שבמימוש שלו טבלאות הדפים תופסות פחות נפח בזיכרון.

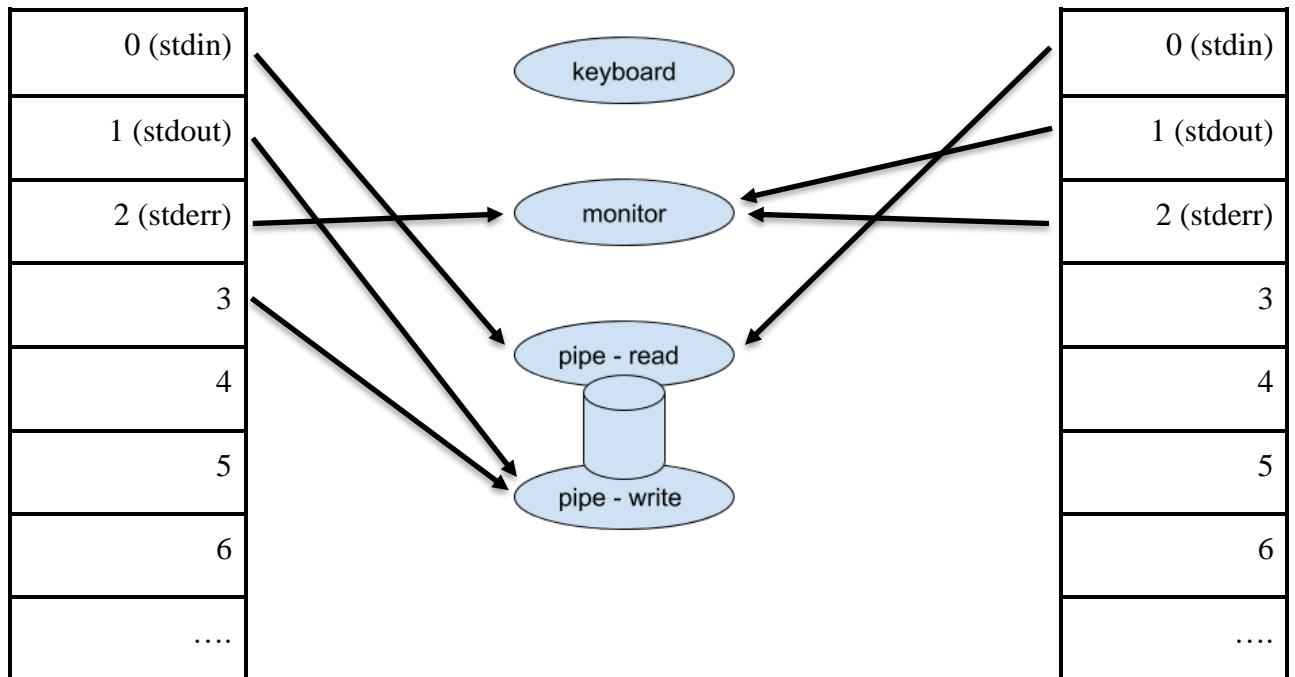
שאלה 2 – סיגנלים:

נתון קטע הקוד הבא:

```
void transfer() { // transfer chars from STDIN to STDOUT
    char c;
    ssize_t ret = 1;
    while ((read(0, &c, 1) > 0) && ret > 0)
        ret = write(1, &c, 1);
    exit(0);
}

int main() {
    int my_pipe[2];
    close(0);
    printf("Hi");
    pipe(my_pipe);
    if (fork() == 0) { // son process
        close(my_pipe[1]);
        transfer();
    }
    close(1);
    dup(my_pipe[1]);
    printf("Bye");
    return 0;
}
```

1. השלימו באמצעות חצים את כל ההצבעות החסרות באיור הבא (למשל חץ מ- stdin ל- keyboard), בהינתן שתהליך האב סיים לבצע את שורה 19 ותהליך הבן סיים לבצע את שורה 15:



2. מה יודפס למסך בסיום ריצת שני התהליכים? (הניחו שקריאות המערכת אינן נכשלות):

- a. Hi
- b. Bye
- c. HiBye
- d. לא יודפס כלום
- e. התהליך לא יסתיים לעולם
- f. לא ניתן לדעת, תלוי בתזמון של התהליכים

נימוק: (תשובה c)

ראשית מודפס "Hi" אל המסך מערוץ הפלט הסטנדרטי. לאחר מכן תהליך האב מדפיס את "Bye" אל ה-pipe write כי זה מה שמחובר אצלו בשלב הזה לstdout. הבן, בתוך פונק' transfer, קורא את pipe-read (מחובר אצלו בstdin) ומדפיס למסך, כי אצל הבן המסך מקושר עדיין לstdout.

בסעיפים הבאים נתבונן בקטע קוד חדש, המשתמש בפונקציה transfer מהסעיף הקודם:

```
int my_pipe[2][2];
void plumber(int fd) {
    close(fd);
    dup(my_pipe[1][fd]);
    close(my_pipe[1][0]);
    close(my_pipe[1][1]);
    transfer();
}

int main() {
    close(0);
    printf("Hi");
    close(1);
    pipe(my_pipe[0]);
    pipe(my_pipe[1]);

    if (fork() == 0) { // son 1
        plumber(1);
    }
    if (fork() == 0) { // son 2
        plumber(0);
    }
    printf("Bye");
    return 0;
}
```


3. מה יודפס למסך כאשר תהליך האב יסיים לרוץ? (הניחו שקריאות המערכת אינן נכשלות) רמז: שרטטו דיאגרמה של טבלאות הקבצים כפי שראיתם בסעיף 1.

Hi .a

Bye .b

HiBye .c

ByeHi .d

e. לא יודפס כלום

f. לא ניתן לדעת, תלוי בתזמון של התהליכים...

נימוק: (תשובה a)

בתחילת התהליך האב מדפיס למסך "Hi". לאחר מכן הוא מדפיס "Bye" לתוך `my_pipe[0][1]` ומסיים את ריצתו. הבן הראשון קורא את ה"Bye" מתוך `my_pipe[0][0]` וכותב אותו לתוך `my_pipe[1][1]`, ואילו הבן השני קורא את ה"Bye" מתוך `my_pipe[1][0]` (מתאפשר כי הבן הראשון כתב לצינור הזה) וכותב אותו לתוך `my_pipe[0][0]` ובעצם נוצרה לנו לולאה אינסופית של כתיבה וקריאה ביניהם שאף פעם לא מודפסת למסך.

אפולו, אל האור, שמע שסטודנטים רבים בקורס עובדים עכשיו על תרגיל הבית במקום ליהנות בחוץ בשמש. בתגובה נזעמת, אפולו הכל יכול התחבר לשרת הפקולטה והריץ את התוכנית הנ"ל N פעמים באופן סדרתי (דוגמה ב-bash, כאשר a.out הוא קובץ ההרצה של התכנית הנ"ל):

```
>> for i in {1..N}; do ./a.out;
```

4. אחרי שהלולאה הסתיימה, נשארו במערכת 0 או יותר תהליכים חדשים. מה המספר המינימלי של סיגנלים שצריך לשלוח באמצעות kill על מנת להרוג את כל התהליכים החדשים שסנטה יצר?

0 .a

1 .b

N .c

N/2 .d

2N .e

f. לא ניתן לדעת, תלוי בתזמון של התהליכים...

נימוק: (תשובה c)

תהליך האב בכל איטרציה מסיים את ריצתו ולכן אינו צריך סיגנל kill שיהרוג אותו. לעומת זאת כפי שהסברנו בסעיף הקודם 2 בניו נמצאים בלולאה אינסופית. לכן אם נהרוג את אחד מהם, השני יצא מהלולאה while שבו היה וגם כן יסיים את ריצתו. זאת בגלל שהם כותבים תמיד Bye אחד לשני וקוראים את Bye שהוא השני כתב להם. ברגע שהרגנו את אחד הבנים לשני לא יהיה Bye חדש לקרוא והוא יסיים את ריצתו.

משמע מספיק לשלוח N סיגנלים של kill (בכל איטרציה לאחד הבנים) כדי להרוג את כל התהליכים.

5. מה תהיה התשובה עבור הסעיף הקודם אם נסיר את שורות 5-6 מהקוד?

a. 0

b. 1

c. N

d. $N/2$

e. $2N$

f. לא ניתן לדעת, תלוי בתזמון של התהליכים...

נימוק: (תשובה c)

גם כאן מספיק לשלוח N סיגנלים אך נבחין כי כעת **חייבים** לשלוח בכל איטרציה את הסיגנל kill לבן מסוג son2. כאשר נשלח סיגנל לבן מסוג son2, אם my_pipe[0] ריק בזמן שליחת הסיגנל, אז son1 לא יקרא ממנו כלום ויסיים את ריצתו, ואילו אם הצינור הנ"ל אינו ריק בזמן שליחת הסיגנל, son1 יקרא אותו ויכתוב לmy_pipe[1] ובסיום כתיבת כל התווים הוא יסיים את ריצתו.

לעומת זאת אם נשלח סיגנל לson1, כאשר son2 ינסה לקרוא מmy_pipe[1] בין אם הוא ריק ובין אם היה בו תווים וסיים לכתוב אותם לmy_pipe[0], הוא לא יסיים את ריצתו לעולם כי עדיין יש כותבים לצינור my_pipe[1] (הוא בעצמו..)