

Operating Systems – 234123

Homework Exercise 4 – Dry

Presented by:

Shiran saada

301731998

Yair Shachar

200431260

Emails:

shiransaada@campus.technion.ac.il

syairsha@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 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:

מכיוון שגודל כתובת זכרון כעת היא ברוחב 40 ביט, וגודל דף הוא 4K נצטרך 12 ביטים עבור הoffset. כעת בכניסה בטבלת הדפים נצטרך 12 ביטים עבור הדגלים ו- $28 = 40 - 12$ ביטים עבור מספר המסגרת – סה"כ 40 ביטים. מכיוון שגודל כניסה בטבלת הדפים מעוגל כלפי מעלה לחזרה שלמה של 2, גודל כניסה תהיה 64 ביט.
 לכן בדף יהיה $2^9 = 2^{15} / 2^6 = 4K / 64$ כניסות.
 לכן ברמה התחתונה והשניה נצטרך 9 אינדקסים כל אחת, וברמה הראשונה את ה-2 הנותרים.

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.

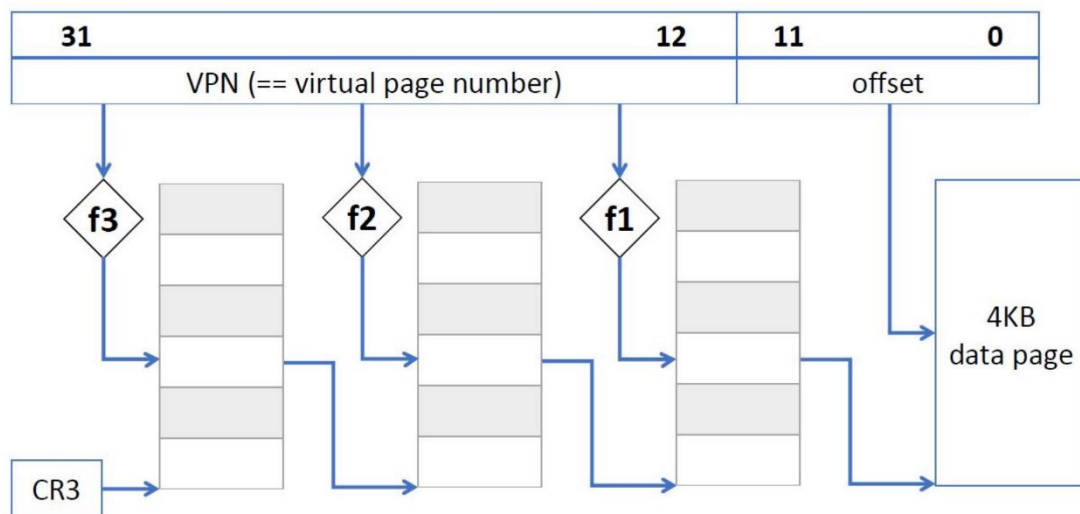
2. מהו הגודל המינימלי האפשרי של כניסה בטבלת הדפים אם לא מעגלים אותו למעלה?

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

נימוק:

כפי שהסברנו בסעיף א' נצטרך 12 ביט עבור הדגלים ו28 עבור מספר המסגרת – סה"כ 40 ביט = 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. $f1(vpn) = (vpn/819)/819$

$$\begin{aligned} \text{d. } f_1(vpn) &= (vpn \% 819) / 819 \\ \text{e. } f_1(vpn) &= (vpn / 819) \% 819 \\ \text{f. } f_1(vpn) &= ((vpn / 819) \% 819) / 819 \end{aligned}$$

נימוק:

מכיוון שכעת כל כניסה של דף היא בגודל 5 בתים, ניתן להכניס $819 = 4KB/8B$ כניסות בדף. מכיוון ש f_1 ממפה מטבלת הדפים אל הזכרון הפיזי, אנו נדרשים למפות את הספרות האחרונות של הכתובת הוירטואלית, שהן יהיו השארית החלוקה של הכתובת ב-819.

4. מהי הפונקציה f_2 ?

$$\begin{aligned} \text{a. } f_1(vpn) &= vpn / 819 \\ \text{b. } f_1(vpn) &= vpn \% 819 \\ \text{c. } f_1(vpn) &= (vpn / 819) / 819 \\ \text{d. } f_1(vpn) &= (vpn \% 819) / 819 \\ \text{e. } f_1(vpn) &= (vpn / 819) \% 819 \\ \text{f. } f_1(vpn) &= ((vpn / 819) \% 819) / 819 \end{aligned}$$

נימוק:

כעת נרצה את הרמה האמצעית של הכתובת. נמצא אותה ע"י חלוקת המספר ב-819 (בצורה זו בעצם אנו מתעלמים מהספרות שאותן שלפנו ע"י f_1) ולאחר החלוקה נמצא את השארית של המספר בחלוקה ב-819

5. מהי הפונקציה f_3 ?

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:

כעת נרצה למפות את הספרות התחתונות ביותר של המספר המתאר את כתובת המסגרת, בכדי לעשות זאת נחלק פעמיים ב819.

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:

מכיוון שגודל כל כניסה בטבלת הדפים קטן יותר (40ביט לאומת 64ביט) ניתן להכניס מספר גדול יותר של כניסות בדף (819 לאומת 512) לכן נוכל למפות את אותו אזור הזכרון כמות קטנה יותר של דפים וכך לחסוך במקום.

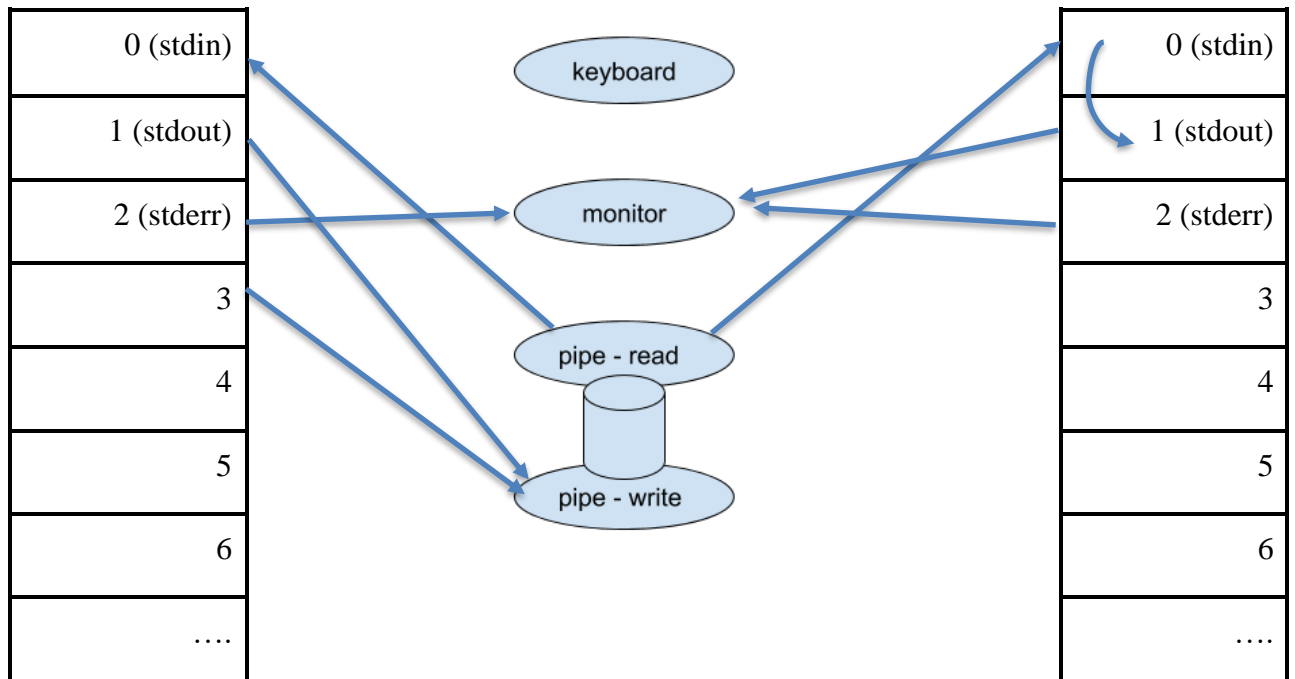
שאלה 2 – סיגנלים:

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:

יודפס תחילה HI ע"י האב ולתוך הצינור שמחובר לstdout יודפס Bye, הבן יעביר את התווים מהצינור אל המסך, שמחובר 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:

תחילה ידפיס האב "Hi" למסך, עוד לפני יצירת הבנים. לאחר מכן ידפיס האב "Bye" אל תוך הצינור שמחובר לבנים ויסיים את פעולתו, הבנים יעבירו ביניהם את ההדפסה "Bye" לנצח ולכן לא יודפס למסך וישאר רק "Hi"

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:

עבור כל שני בנים המעבירים ביניהם את ההודעה "BYE" מספיק לשלוח סיגנל אחד בכדי להרוג אותם- לאחר ששלחנו סיגנל לאחד הבנים, הבן השני יקבל EOF במידה והצינור ריק, אחרת הוא ינסה לשלוח תו דרך צינור שאין לו קוראים ולכן יקבל SIGPIPE וימות. סה"כ יש לנו N תהליכים ולכל תהליך 2 בנים- סה"כ 2N בנים, ולכן נדרש N' סיגנלים.

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:

התשובה תשאר N פעמים אך במקרה זה יש חשיבות למי שיקבל את הסיגנל - נדרש לשלוח סיגנל רק ל son 2 כך ש son1 יקרא EOF וימות (בשני מקרי קבלת הסיגנל כי גם במקרה שהצינור אינו ריק כאשר son2 מסתיים, son1 יקרא את כל התווים בצד השני ועדיין יקבל EOF).

במידה ונשלח ל son 1 סיגנל, son2 יכנס להמתנה מכיוון שעדיין יש כותב בצד השני של הצינור.

סה"כ $2N$ בנים ודרוש סיגנל 1 לכל זוג ולכן סה"כ נדרש ל N סיגנלים.