

Operating Systems – 234123

Homework Exercise 4 – Dry

Presented by:

Omer Stoler: 318471356

Adi Arbel: 207919614

Emails:

stoler.omer@campus.technion.ac.il

adi.arbel@campus.technion.ac.il

Teaching Assistant in charge:

Sami Zreik

Assignment Subjects & Relevant Course material

Virtual Memory & IPC

Recitations 7 & 10, Lecture 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
3. The submission should contain the following:
 - a. The first page should contain the details about the submitters - Name, ID number and email address.
 - b. Your answers to the dry part questions.
4. 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 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. Convolved 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://goo.gl/forms/Im4tFXNLq4uznBzM2>

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 הווירטואליות לשדות בתהליך תרגום כתובות (page walk) ?

	index3	index2	index1	offset
א.	2	9	9	12
ב.	2	10	10	10
ג.	1	9	10	12
ד.	4	9	9	10
ה.	2	10	10	12
ו.	2	9	9	10

נימוק:

התשובה היא א'

כיוון שהזיכרון הפיסי ממפה מרחב כתובות של 2^{40} אנו צריכים לשמור כתובות ב-PTE שגודלן 40 ביט לפחות. לאור הדרישה לעגל גודל כניסות בטבלה לחזקה שלמה של 2 עלינו לשמור בפועל 64 ביט בכל כניסה בטבלה. מכאן שבכל frame בגודל 4KB אנו יכולים לשמור 512 כניסות בגודל של 8 בתים ועל כן ניתן לאנדקס אותם בשימוש ב-9 ביטים בלבד. בנוסף כיוון שכל מסגרת היא בגודל 4KB יש לשמור OFFSET של 12 ביטים כדי למפות כל תוכן של מסגרת. לכן לא נותר אלא לחלק ל: 9 9 12 והיתר שנותרו מתוך 32 ביט הינם 2 ביטים עבור הרמה העליונה ביותר בטבלאות הדפים.

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.

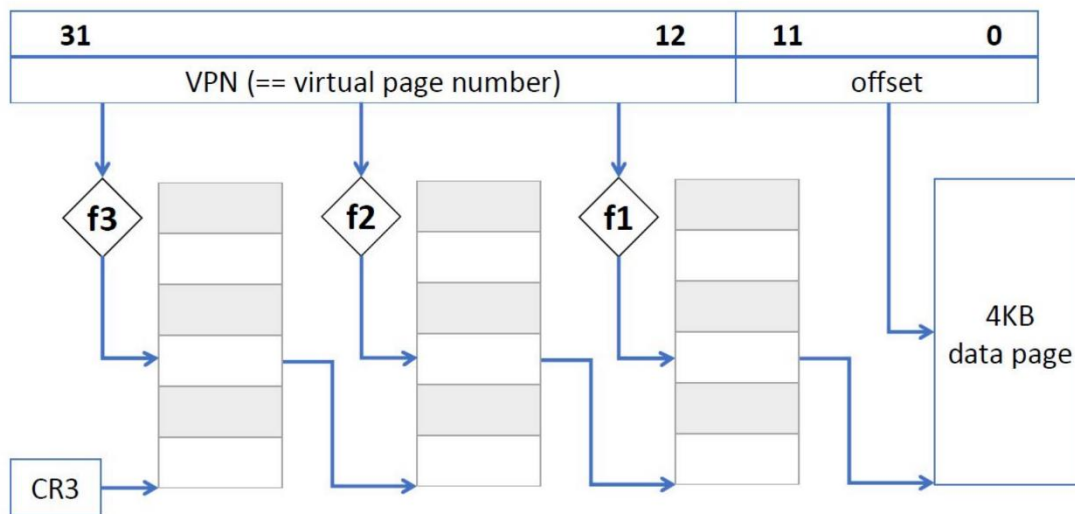
2. מהו הגודל המינימלי האפשרי של כניסה בטבלת הדפים אם לא מעגלים אותו למעלה?

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

נימוק:

התשובה היא ג' - כיוון ש40 ביטים נכנסים בדיוק בחלוקות של 8, נוכל כל כתובת לשמור 5 בתים בלבד.

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:

$$1024/819=1$$

$$1024\%819=205$$

תיאור כללי של הדרך אל התשובה : אנחנו יודעים כי בכל כניסה בטבלה מאוחסנים 5 בתים כאשר כל מסגרת היא בגודל 4096 לכן נכנסות לכל היותר $\text{floor}\left(\frac{4096}{5}\right) = 819$ כניסות. מכאן שיש למפות את המסגרות באופן היררכי, כך שכל רמה בהיררכיה לא מקבלת יותר מ-819 ערכים שונים. על כן עלינו להעביר את 20 הביטים שברשותנו לייצוג בבסיס המתאים – 819 ולכך ייעשה שימוש בפעולות div/mod לחילוף הערך המתאים לכל רמה בבסיס זה. באנלוגיה לבסיס 10 : אם ניקח את המספר 123 ונרצה לחלץ ממנו כל ספרה בבסיס 10 נבצע :

$$123 \% 10 = 3$$

$$(123/10) \% 10 = 2$$

$$((123/10)/10) \% 10 = 1$$

כאשר במקרה הנתון המודולו האחרון לא היה נחוץ בגלל גודל המספר. אותה פעולה נעשית באופן כללי לכל בסיס כאשר את 10 מחליף במקרה שלנו 819.

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:

התשובה היא ב' – כך מחלצים את ספרת ה"אחדות" של בסיס , בפרט 819.

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:

התשובה היא ה' – כך מחלצים את ספרת ה"עשרות" של בסיס , בפרט 819.

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:

התשובה היא ג' – כך מחלצים את ספרת ה"מאות" של בסיס , בפרט 819. בגלל שספרת המאות של המספר

$$2^{20} \text{ בבסיס } 819 \text{ הינה } 1 \text{ floor}\left(\frac{2^{20}}{819^2}\right) \text{ אזי המודולו אותו רשמנו לעיל אינו נחוץ.}$$

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:

התשובה היא ד' – נשים לב כי החסכון העיקרי הוא תחזוק פחות מסגרות למיפוי הזיכרון על ידי דחיסה של כניסות ל-40 ביטים במקום 64. על כן היתרון צריך לבוא בחסכון בזיכרון. מרחב הזיכרון שניתן למפות בהכרח נותר זהה כי נותרנו עם אותו מספר ביטים בכתובת הווירטואלית, ולמעשה מנענו פרגמנטציה פנימית – כלומר חסכון בזכרון שהוקצה ולא נעשה בו שימוש. מכאן שאף תשובה אחרת לא תתאים וכעת נראה את החסכון שהתקבל – נתבונן בחסכון שהתקבל בכל הרמות מבחינת כמות הטבלאות שנחוצה:

עדין:

הרמות העליונות בעץ נותנות יחד מספר טבלאות $1 + 4 + 4 \cdot 512 = 2053$ מסגרות המתחזקות את היררכיית הדפים במקרה זה.

שוקי:

הרמות העליונות לא מאוזנות בעץ ההיררכיה זאת כיוון שמספר המסגרות 2^{20} לא מתחלק ב- 819^2 . על אף זאת, נוכל לקבל את מספר המסגרות תוך התחשבות בכך שאינן ממלאות את כל הטבלה

$$1 + 2 + (o_1 \cdot 819 + o_2 \cdot 819) \approx (o_1 + o_2)819$$

כאשר o מייצג בכל טבלה מבין השתיים את מידת הכניסות שעבורן מוקצה טבלה. לאור העובדה שאין צורך במילוי שתי הטבלאות להקצאה של 2^{20} מסגרות סה"כ חלק לא ידרשו להקצאה – נתח זה יהיה כמות המסגרות שנותרה אחרי שמילאנו את הטבלה העליונה חלקי הקיבולת של הטבלה התחתונה כולה

$$\dots = 0.5632 = \frac{2^{20} - 819^2}{819^2} \quad \text{לכן נקבל שמספר הטבלאות הנחוצות בזכרון הינו}$$

$$\approx \text{ceil}((1 + 0.5632)819) = 1281$$

כלומר חסכנו בהיקף נרחב.

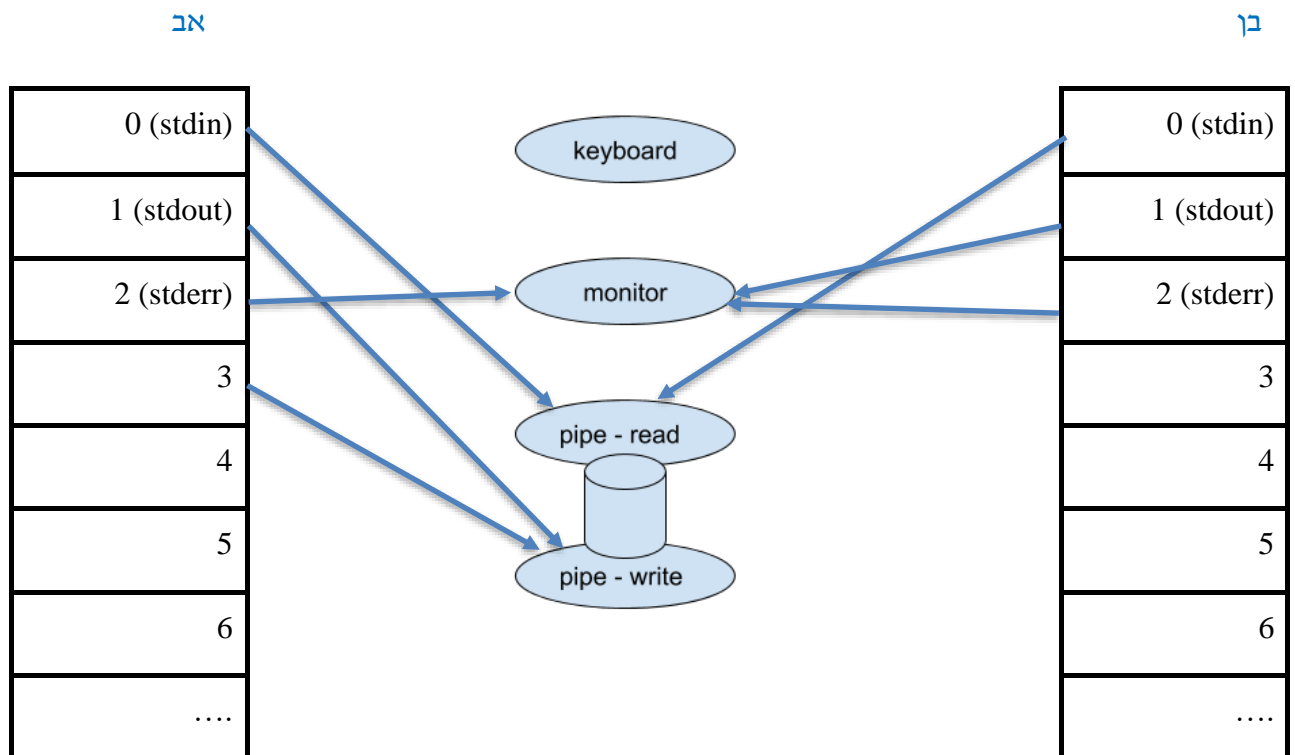
שאלה 2 – סיגנלים:

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:



בעקבות הרצת הקוד :

עד הפיצול נסגר STDIN ונפתח PIPE שערך הקלט שלו תפס את מקום הSTDIN והפלט תפס את המקום הפנוי הבא שהוא 3. לאחר מכן נעשה פיצול וכל תהליך שינה טבלה זו בנפרד לאחר שכפולה

- האב – סגר את ערוץ הפלט הסטנדרטי 1 והעתיק לתוכו את ערוץ הפלט של הPIPE לכן נקבל שכפול בהצבעה זו. ערוץ השגיאות הסטנדרטי לא שונה בכלל ולכן מצביע למסך
- הבן – סגר את ערוץ הפלט של הPIPE ולכן רשומה 3 שלו ריקה. בנוסף עוד לא הגיע לסגירת ערוץ הפלט הסטנדרטי ולכן הוא עדיין מצביע למסך. כמו כן לא חל שינוי בערוץ השגיאות הסטנדרטי.

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:

לאור העובדה שקריאה היא פעולה חוסמת, ושיש את האב שמצביע על הקצה הכותב של PIPE, ימתין הבן עד שיסיים האב לכתוב מתוך PRINTF את כל המילה, ולאחר מכן יצליח לכתוב כל אחת מהן בתורו לערוץ הפלט הסטנדרטי. כיוון שהכתיבה לPIPE היא באופן סדרתי הטקסט יגיע בסדר הנכון. לאחר כל הכתיבות הבן ייחסם לאור READ, וימתין עד שהאב יסיים את התהליך (כיוון שלא סוגר באופן יזום את PIPE) ולאחריו READ כבר לא תחסום אותו ויסיימו שניהם בהצלחה עם הדפסה תקינה של HiBye

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

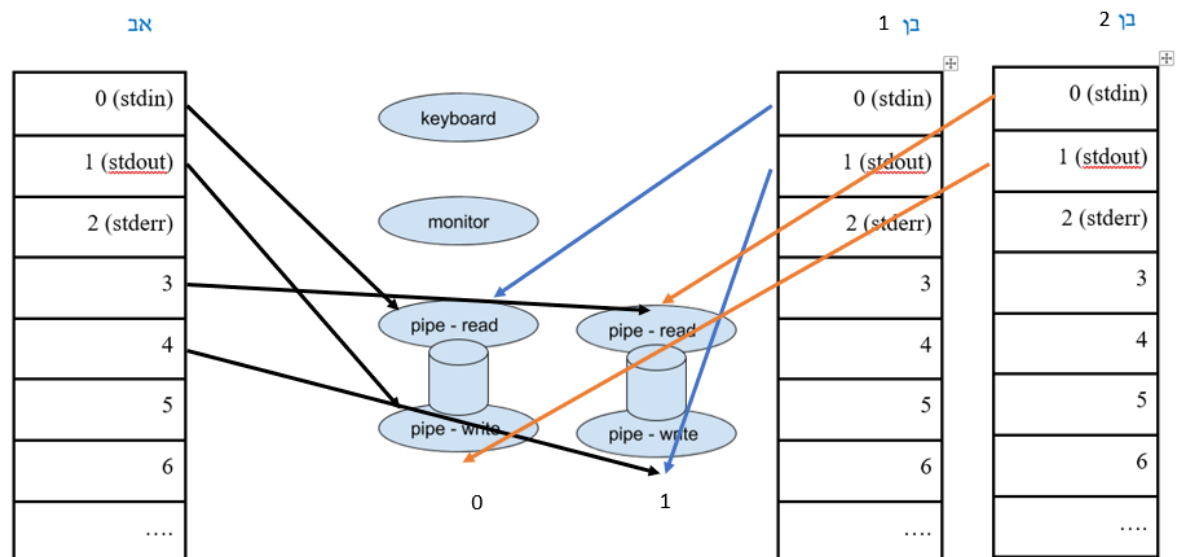
HiBye .c

ByeHi .d

.e לא יודפס כלום

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

נימוק:



Hi בהכרח מודפס כיוון שנעשה לפני שינוי הפלט הסטנדרטי להיות PIPE במקום המסך. כמו כן ציירנו את מצב ההצבעות בטבלת FD (ללא STDERR שמצביע למסכים כדי לחסוך בחיצים). לפי הצבעות אלה האב ידפיס את ה-BYE ל-PIPE ויסיים את ריצתו ולאחריה ניתן לראות כי יש מעגליות בין שני הצינורות שנוצרו – בן 1 כותב ל-PIPE 1 וקורא מ-0 והבן השני כותב ל-0 וקורא מ-1 לכן הם יתמסרו במידע עד אין קץ ולא נראה דבר על המסך כיוון ששניהם מנותקים מהמוניטור בטבלת ההצבעות לקבצים.

אפולו, אל האור, שמע שסטודנטים רבים בקורס עובדים עבשיו 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:

N הרצות יגרמו לN זוגות תהליכים להתמסר בPIPE עד אינסוף. כיוון שההרצות לא תלויות אחת בשניה עלינו לשלוח לפחות KILL אחד לכל תהליך שנולד מההרצה, אלא שהריגת תהליך אחד בזוג תגרום לקריאה שמתבצעת על ידי השני מהם להחזיר ערך שגיאה מכיוון שאין מי שכותב בPIPE יותר מרגע שהאב סיים וגם הבן שמת. אם כן יש לשלוח לכל זוג KILL כלומר N סה"כ.

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

0 .a

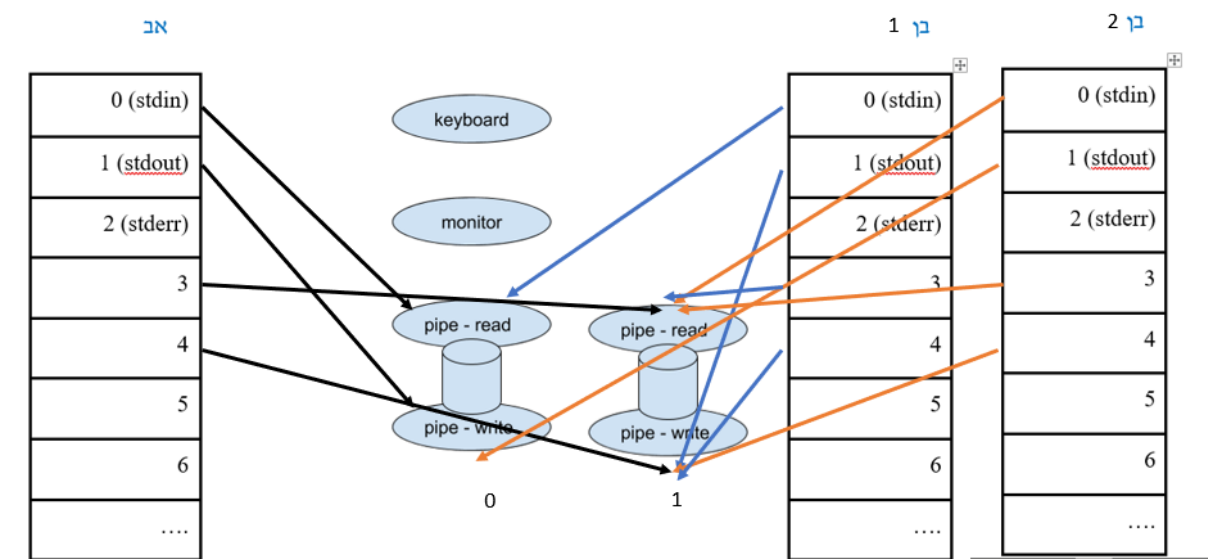
1 .b

N .c

N/2 .d

2N .e

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



נימוק:

גם כאן N הרצות יגרמו לזוגות אלא שכעת אין סימטריה בתפקידו של כל בן. לעיל מוצגת טבלת ההצבעות (שוב ללא STDERR לשם נוחות) ונקבל את המצב הבא

- אם בן 1 נהרג – נותרים עם בן 2 שקורא קלט סטנדרטי מPIPE שהוא מצביע אליו ולכן ייחסם על ידי READ לנצח ויש להורגו גם
 - אם בן 2 נהרג – בן 1 יקרא קלט סטנדרטי מPIPE שלא מצביעים עליו ולכן יסיים.
- מכאן שנוכל לשלוח באופן מינימלי רק N הריגות, אחת לכל בן 2 של כל אחת מן הריצות וכך הזוג ייהרג.