Homework 4 Dry

**Due Date: 15/01/2018 23:30**

**Teaching assistants in charge:**

* **Sami Zreik (**[**samizr@cs.technion.ac.il**](mailto:samizr@cs.technion.ac.il)**)**
* **Omer Glick (**[**omer.glick@cs.technion.ac.il**](mailto:omer.glick@cs.technion.ac.il)**)**

**Important:** the Q&A for the exercise will take place at a public forum Piazza only. 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

Only the TA in charge can authorize postponements. In case you need a postponement, please fill out the following form: <https://goo.gl/forms/nMCWQvYeKEKm4pq82>

**Dry part submission instructions:**

1. Please submit the dry part to the electronic submission of the dry part on the course website.
2. The dry part submission must contain a single dry.pdf file containing 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.
3. Only typed submissions will be accepted. Scanned handwritten submissions will not be accepted.
4. Only PDF format will be accepted.
5. You do not need to submit anything in the course cell.
6. When you submit, **retain your confirmation code and a copy of the PDF**, in case of technical failure. It is **the only valid proof** of your submission.
7. IMPORTANT: Please write your answers in a different **COLOR** (not black).

Question 1 - Pipes & I/O:

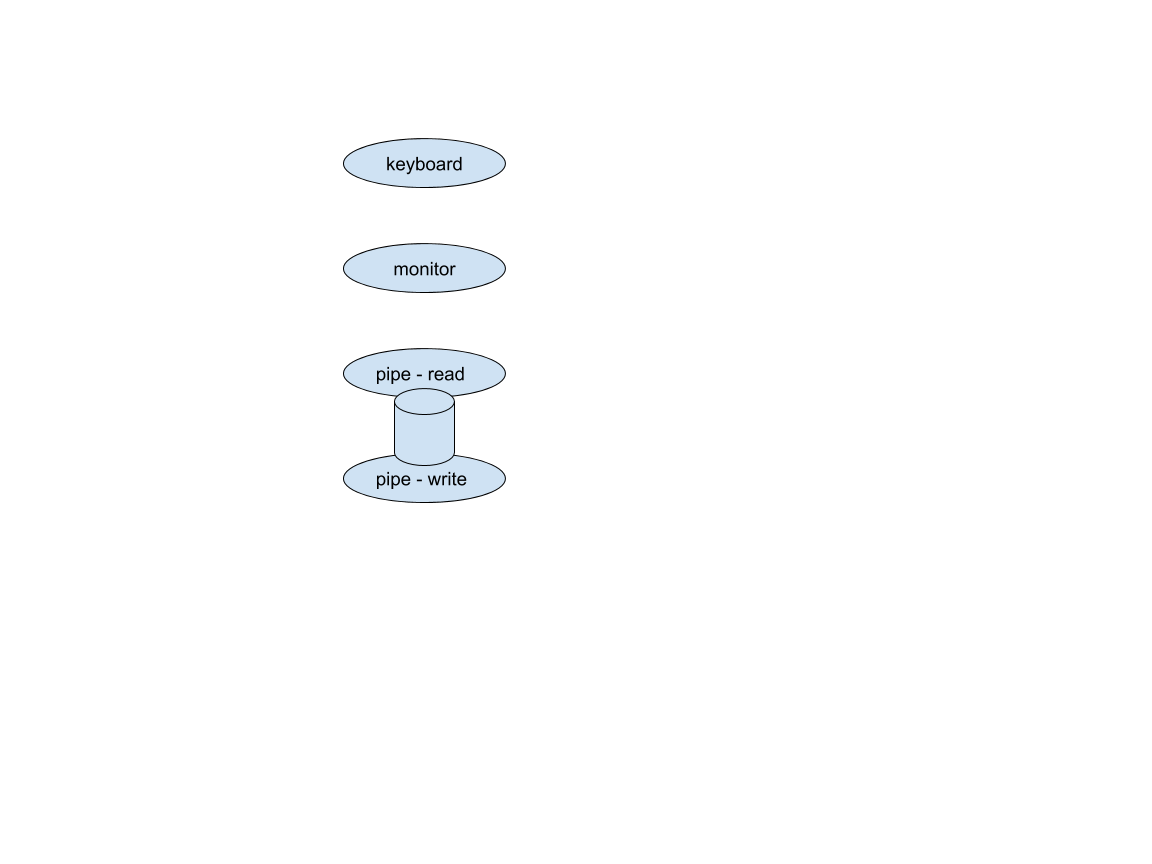
נתון קטע הקוד הבא:

|  |
| --- |
| 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:



|  |  |  |
| --- | --- | --- |
| 0 (stdin) |  | 0 (stdin) |
| 1 (stdout) |  | 1 (stdout) |
| 2 (stderr) |  | 2 (stderr) |
| 3 |  | 3 |
| 4 |  | 4 |
| 5 |  | 5 |
| 6 |  | 6 |
| …. |  | …. |



1. מה יודפס למסך בסיום ריצת שני התהליכים? (הניחו שקריאות המערכת אינן נכשלות):  
   1. Hi
   2. Bye
   3. HiBye
   4. לא יודפס כלום
   5. התהליך לא יסתיים לעולם
   6. לא ניתן לדעת, תלוי בתזמון של התהליכים

נימוק:

הHi יודפס שכן אין כל מניעה לכך. לאחר מכן, במידה שהבן ירוץ קודם, לא יהיה תוכן בpipe שכן האבא עוד לא ביצע את הכתיבה אליו, לכן הוא ייחסם עד שהאבא יכניס את התוכן ולאחריו הבן יוכל להדפיס אותו. אחרת, האבא יכניס את המידע לpipe והבן שינסה להדפיס את המידע הזה למסך יצליח. ובשני המקרים יודפס בסוף 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;  } |
|  |

1. 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:

מכיוון שכבר בהתחלה, עוד טרם הfork()ים אנחנו סוגרים את הstdout, דרכו אנחנו מדפיסים למסך, כל תהליך ישכפל את הDiscriptor Table של האב ללא התקן זה. לפיכך, אף אחד מן הבנים לא יוכל לבצע transfer למסך שכן stdout לא יהיה בDT שלהם. בשורה התחתונה – אנחנו נדפיס את Hi בלבד. כמו כן, הבנים מעבירים את הקלט בין הpipים שלהם במעגליות.

סנטה קלאוס שמע שסטודנטים רבים בקורס עבדו במהלך הכריסמס על תרגיל הבית, ואפילו נהנו ממנו יותר מאשר במסיבת הסילבסטר של הטכניון. בתגובה נזעמת, סנטה התחבר לשרת הפקולטה והריץ את התוכנית הנ"ל N פעמים באופן סדרתי (דוגמה ב-bash ,כאשר out.a הוא קובץ ההרצה של התכנית הנ"ל):

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

1. 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?

* 1. 0
  2. 1
  3. N
  4. N/2
  5. 2N
  6. לא ניתן לדעת, תלוי בתזמון של התהליכים…

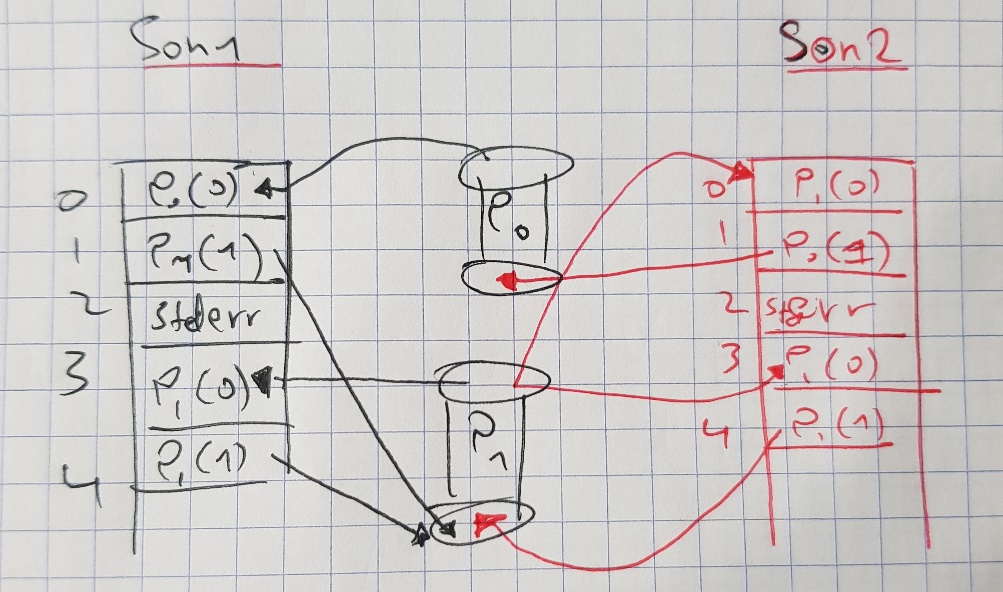
נימוק:

בשאלה הקודמת ראינו שיש מעגל אינסופי בין הpipים של שני הבנים. לפיכך, עבור הרצת N התוכניות, נקבל N זוגות תהליכים שתלויים אחד בשני ויוצרים N מעגלים אינסופיים. לפיכך, מספיק להרוג את אחד הבנים שנמצאים במעגל אינסופי (לכל N הזוגות) כדי להרוג את כל התהליכים.   
נשים לב כי ייתכנו שני מצבים כאשר נהרוג את אחד הבנים: אם בPIPE יש מידע שאותו נרצה לכתוב לבן שהרגנו, אזי נקבל סינגל SIGPIPE . אחרת, אם אין מידע בPIPE, הבן שלא נהרג ינסה לקרוא מידע ממנו, יראה כי אין כותבים ולכן יבצע exit(0).

1. מה תהיה התשובה עבור הסעיף הקודם אם נסיר את שורות 5-6 מהקוד?
2. 0
3. 1
4. N
5. N/2
6. 2N
7. לא ניתן לדעת, תלוי בתזמון של התהליכים…

נימוק:

נשרטט את שני הבנים ונראה כי **רק** הריגת הבן השני תוביל לסיום מוצלח של כל התהליכים:

  
במצב בו בן 1 הסתיים לרוץ ויש ב1P נתונים, בן2 יקרא אותם אבל 1P לא יסגר שכן עדיין יש מצביעים לכתיבה בו, אותם המצביעים שכעת לא סגרנו בשורות 5-6, כלומר בן2 ייחסם לנצח. אנחנו מוכרחים להרוג את בן 2 אם כך.  
נראה כי סגירת בן 2 מספיקה:  
במקרה שבו 0P מכיל תוכן – בן 1 יקרא אותו, יכתוב ל1P ויצא, שכן אין יותר מצביעים לכתיבה (סגרנו את בן2).  
במקרה שבו 1P לא מכיל תוכן – פשוט נצא, שכן ה0P ריק וגם לא מכיל מצביעים.  
  
לפיכך, התשובה היא עדיין N אם כי הפעם רק הריגת בן 2 תוביל להריגת כל התהליכים.

Question 2 – Interrupts, signals

1. סערת פסיקות (storm interrupt) הוא כינוי לתופעה המתרחשת כאשר קצב הגעת הפסיקות גבוה מאוד מה הבעיה במצב זה וכיצד ניתן להתמודד איתה? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. נתון כרטיס רשת העובד בקצב של 20 Gbit/sec ושולח ומקבל חבילות (packets) בגודל קבוע של 4096 bit. נניח כי מערכת ההפעלה רוצה לעבוד עם הכרטיס בשיטת polling.
   1. מדוע לא כדאי לקבוע את תדר ביצוע ה- polling ל- 0.5 GHz? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   2. מדוע לא כדאי לקבוע את תדר ביצוע ה- polling ל- 5 KHz? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. בתרגול ראינו כי בפונקציה schedule לפני שניגשים לבצע פעולות על ה-runqueue משתמשים בפקודה irq\_lock\_spin אשר תופסת את ה-spinlock של ה-runqueue וחוסמת את הפסיקות במעבד הנוכחי.
   1. מדוע יש צורך לחסום פסיקות ולא ניתן להסתפק בתפיסת המנעול? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   2. באיזה בעיית סנכרון היינו נתקלים אם לא היינו חוסמים את הפסיקות? יש לתאר מצב בו מתרחשת בעיה זו. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. נתונה מערכת בעלת שני מעבדים B,A אשר מריצים שני תהליכים PB,PA בהתאמה. ברגע מסוים PA שולח סיגנל USR1 אל תהליך PB. הניחו כי מערכת ההפעלה אינה מבצעת החלפת הקשר ושהפסיקות והסיגנלים לא חסומים.

a. מה פרק הזמן המקסימלי בין הרגע בו נשלח הסיגנל לרגע בו PB יתחיל לטפל בו? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b. תן דוגמה למצב בו PB יתחיל לטפל בסיגנל לאחר פרק זמן קצר יותר. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. תארו את ההבדל בהתנהגות של הפקודה iret בעת חזרה מפסיקה מקוננת לעומת חזרה מפסיקה לא מקוננת. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_