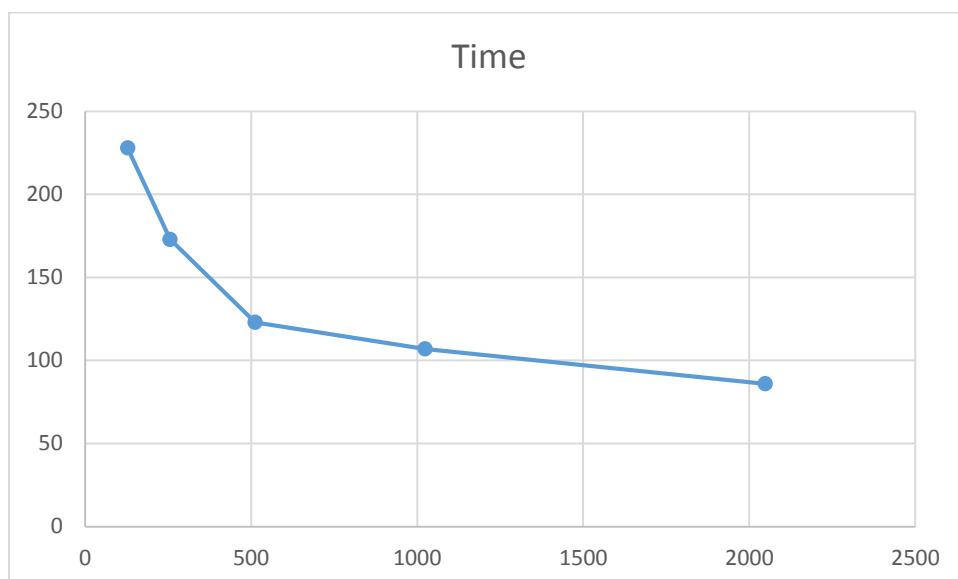


## שאלה 2

(1)



(2) מכיוון שככל שגודל הבאפר גדל בכל הרצה שאנו מבצעים אז ברור מדוע זמן הפעולה מתקצר, זה קורה מכיוון שבכל פעם שהבאפר מתמלא הוא צריך לגשת אל הזיכרון ולרוקנו לשם. פעולה זו לוקחת זמן ולכן ככל שהבאפר קטן הוא מבצע יותר גישות לזיכרון שמשפיע באופן ישיר על זמן הריצה (כמוכן שיש חשיבות גדולה לגודל הטקסט בעניין זה).

(3) באופן היפותטי, כל ביצוע של פעולת הדפסה זו בעצם בקשה ממערכת הפעלה שלוקחת זמן מה, לכן ככל שהבאפר יותר קטן אז הוא ייגש יותר לזיכרון (מסעיף קודם) ועכשיו, בנוסף לכך, הוא גם מבצע על כל פעם כזו פעולת הדפסה ולכן הפערים בין הרצה להרצה מהסעיף הקודם רק יגדלו. כלומר, כל הרצה תגדל באופן יחסי לכמות הפעמים שהבאפר מתמלא.

## שאלה 3

- (1) לא נכון, מכיוון שכאשר תכנית מסוימת מבצעת פעולת הדפסה, בעצם מה שקורה זה שהיא ניגשת למערכת הפעלה כדי לבצע זאת (כמו שנאמר בסעיף ג' בשאלה הקודמת).
- (2) לא נכון, לחיצה על כפתור המקלדת כמו שהוא זה interrupt.
- (3) לא נכון, בעת היווצרות של interrupt נשלחת ממנו התראה לCPU ולא להפך.
- (4) לא נכון, באופן הפוך לגמרי דווקא בuser mode ניתן לבצע system calls.
- (5) לא נכון, Command Shell אכן מהווה חלק ממערכת ההפעלה אך עובד בUser Mode.
- (6) נכון, כאשר מערכת ההפעלה נמצאת בעומסים גדולים מאוד יש לה רכיב שכאשר הוא דולק (=1) היא בעצם לא מתייחסת ל interrupts.
- (7) לא נכון, דווקא שימוש בVM תגרום לאיטיות יותר גדולה מאשר מהמערכת המקורית שכן ה VM עוד עובר בשכבות נוספות שהמערכת המקורית לא.
- (8) לא נכון, אנו לא חייבים להיות ב kernel mode בשביל לגשת לכונן מסוים, אנו יכולים לשלוח System call למערכת ההפעלה והיא גם תיגש לכונן.
- (9) לא נכון, ככל שישנן יותר בקשות System calls דווקא נכביד על המעבד שלנו שכן רצות יותר תכניות ומן הסתם תאט את המערכת (ייתכן שלא נרגיש האטה אך בטוח שזה לא ישפר את מהירות המערכת).
- (10) לא נכון, תכניות הן היחידות שיכולות לבצע System calls. רכיבים חיצוניים משתמשים ב interrupts כדי לגשת למערכת (שכן המערכת לא יכולה "לצפות" קריאות של רכיבים חיצוניים שמושפעים ע"י User).