

שם: עידן פרייברג  
ת.ז: 302939178

## מערכות הפעלה – תרגיל מס' 1

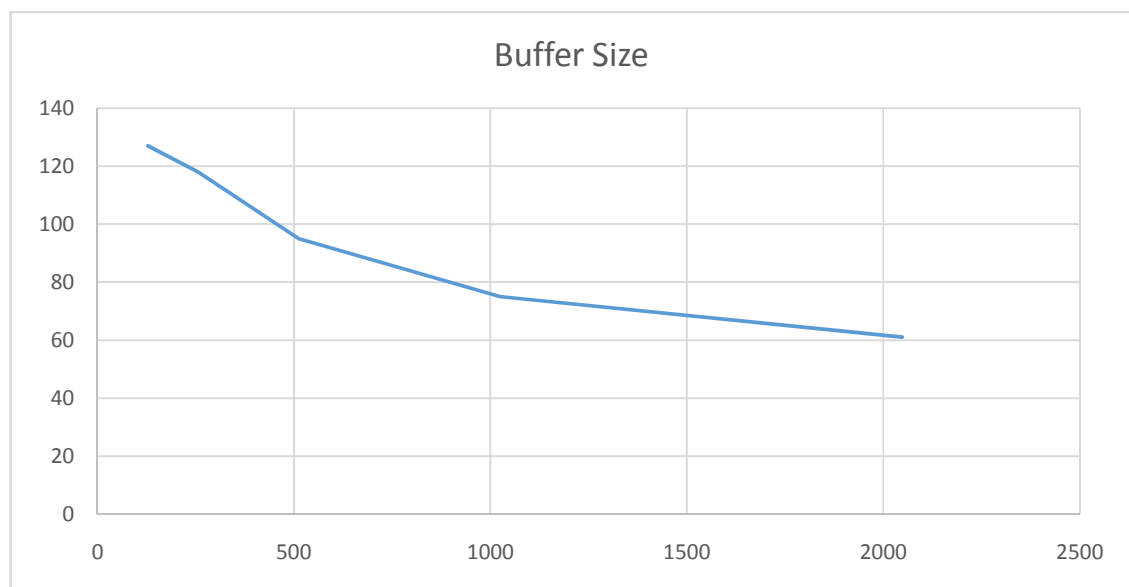
### חלק 1

מצורף קובץ TimeTest.java

### חלק 2

1. להלן תוצאות הבדיקה

BufferSize(bytes)	128	256	512	1024	2048
RunningTime(ms)	127	118	95	75	61



2. ככל שאנו קוראים כמות גדולה של מידע פר Chunk כמות שהאנשים שנקרא תהיה קטנה יותר, וכן יתבצעו פחות קריאות לפונקציית מ"ה read() (העטופה), לכן זמן הריצה יקטן בהתאם. מסיבה הזו הגרף לא יכול להיות קו ישר מקביל לציר X (כלומר זמן קבוע).

3. כן זמן הריצה שישתנה משמעותית (יגדל) כיוון שהדפסה למסך זוהי עוד קריאה למערכת ההפעלה שהינה יקרה / כבדה ולכן תגרום לזמני הריצה לעלות.

### חלק 3

1. לא נכון, הדפסה למסך דורשת להשתמש בsyscall על מנת להעביר את המידע להתקן המסך. מערכת ההפעלה היא זו שאחראית לדבר עם ההתקנים.
2. לא נכון, בדרך כלל הקשת מקלדת גורמים לHardware interrupts בכל כמה זמן ומערכת ההפעלה תטפל בהם.
3. לא נכון, להיפך Interrupts ישלחו בדרך כלל מהתקני חומרה כמו המקלדת למעבד כמו בדוגמא בשאלה 2. ישנם גם Software interrupts שיכולים להיגרם על ידי תוכנה.

4. לא נכון, Syscalls משמשות מעטפת שמספקת מערכת ההפעלה שבעזרתן ניתן לבצע עבודה מול החומרה ללא צורך לדיבור ישיר איתה. למשל כתיבה לדיסק, הדפסה וכד'. לכן אפליקציות שרצות בUSER אמורות ויכולות להשתמש בSyscalls.
5. לא נכון. האפליקציות שבאות עם מ"ה באופן מובנה וגם OFFICE ואפליקציות אחרות שאנחנו מתקינים רצות בUSERMODE. רק דרייברים (שמיועדים לדיבור עם חומרה מסוימת) וקוד פנימי של מערכת ההפעלה רץ ברמת KERNEL.
6. נכון. ניתן ברמת KERNEL לבטל INTERRUPTS לצורך נעילות של משאבים אשר יכולים להיקרא מ-context של interrupts שונים.
7. לא נכון, מכיוון שה VM צורך שירות מהמעבד של ה HOST שעליו הוא רץ וכל הבקשות יצטרכו לעבור קודם דרך ה HOST באמצעות יכולות וירטואליזציה שקיימות ולכן זה לא בהכרח נכון.
8. לא נכון, התוכנה תרוץ בUSER ותוכל להשתמש בSYSCALLים שונים לצורך נגישות ודיבור עם חומרה.
9. לא נכון, קריאות syscall הינן דבר יקר מכיוון שהן יכולות לגרום לcontext switch שלוקח זמן.
10. לא נכון, כל התקשורת של החומרה עם מערכת ההפעלה קורית ברמת KERNEL ולכן אין שימוש בsyscalls מהחומרה.