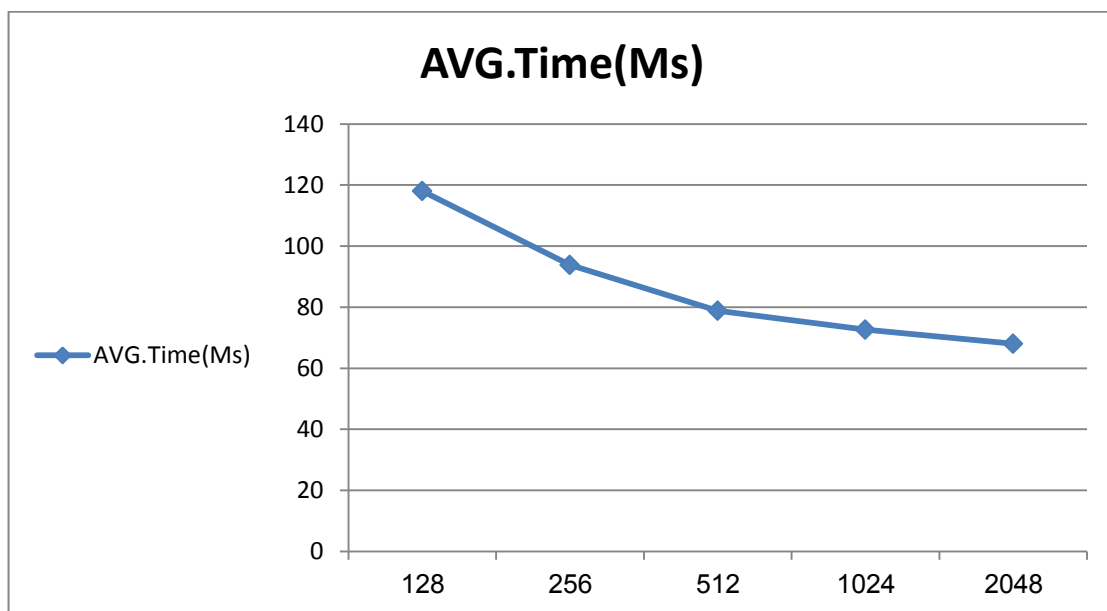


חלק 2:

1.

BufferSize(Byte)	AVG.Time(Ms)	run1	run2	run3	run4	run5
128	118	116	115	139	99	121
256	93.8	106	87	81	88	107
512	78.8	95	63	75	88	73
1024	72.6	65	60	78	82	78
2048	68	60	60	71	76	73



2. הגרף אינו קו ישר המקביל לציר ה-X (קבוע) מכיוון שככל שגודל הבאפר קטן יותר כך אנו נדרשים ליותר System calls :

- אנו קוראים אל הבאפר מקובץ המקור (System call)
- הבאפר מתמלא
- אנו מרוקנים את הבאפר אל קובץ היעד (System call)
- חוזרים על סעיפים א' - ג' עד שסיימנו להעתיק

ככל שהבאפר קטן יותר אנו חוזרים יותר פעמים על הלולאה הנ"ל ולכן מייצרים יותר System calls.

כל System call גוררת contact switch (התהליך אל מול המעבד) אשר עולה לנו בזמן חישוב.

3. בסיטואציה המתוארת זמן הריצה יגדל מכיוון שהוספנו לתוכנה עוד System calls ועל כן עוד זמן חישוב "יתבזבז" על contact switch.  
ככל שהבאפר קטן יותר (יתמלא יותר פעמים) כך זמן הריצה של התוכנה יגדל.

חלק 3:

1. False, הקריאה להדפסה למסך היא עצמה פניה למערכת ההפעלה (System call).

2. False, לחיצה על מקש כלשהו במקלדת תגרום ל-Interrupt ולא ל-System call.
3. False, ההיפך הוא הנכון – Interrupts הם סגנלים הנשלחים ע"י גורם חיצוני (IO devices) אל ה-CPU.
4. False, נכון שדפדפנים רצים ב-User mode אבל עצם העובדה הזאת מונעת מהם גישה ישירה לחומרה ולכן הם חייבים להיעזר ב-System calls.
5. False, גם תוכנות שמגיעות עם מערכת ההפעלה (ואינן מערכת ההפעלה עצמה) רצות ב-user mode.
6. True, מערכת ההפעלה יכולה לבטל Interrupts (disable) על מנת להשיג גישה בלעדית ל-CPU.
7. False, כוח החישוב של כל VM הינו קטן או שווה לכוח החישוב של המכונה האמיתית עליה היא מורצת ולכן עדיף מלכתחילה להריץ את התוכנית על המכונה האמיתית.
8. False, התוכנית תרוץ ב-User mode ותיגש ל-CD-ROM (וכך למידע המאוכסן בו) ע"י ביצוע System calls.
9. False, More system calls → More context switch's → More time waste.
10. False, הם ניגשים למערכת ההפעלה ע"י Interrupts. (System calls מבוצעות ע"י Process's).