

חלק 2

א.



ב. הסיבה לכך שזמן הריצה הולך ויורד, ולא נשאר קבוע, היא שככל שגודל ה- buffer גדול יותר התוכנית תבצע פחות system calls. כאשר המספרים קטנים, כמו 128 או 256 גודל buffer, מבוצעים יותר קריאות למערכת ותהליך זה אורך זמן.

ג. יהיה שינוי משמעותי בין שתי הפעולות. אמנם - גם כתיבה לקובץ וגם כתיבה ל- System יוצרת system call וזה גורם לעיקר העיכוב עד סיום התוכנית. אך במקרה של הדפסה למסך, יתבצע גם Context Switch ומעבר למצב Kernel Mode כדי לגשת לחומרה (למסך), להדפיס שם, ולחזור. ועל כן - פעולה של הדפסה כבדה יותר מאשר פעולה של כתיבה לתוך קובץ.

חלק 3

1. הטענה אינה נכונה, הדפסה ל- system הינה system call המבקשת להדפיס את הטקסט למסך.
2. הטענה אינה נכונה, לחיצה על מקש במקלדת יוצרת Interrupt בלבד, ולא system call. קריאות מערכת נוצרות על ידי processes.
3. הטענה אינה נכונה, interrupts נשלחים ממכשירים חיצוניים ל- CPU, על מנת לבצע פעולה או להודיע על תקלה, כדי שהמחשב יגיב בהתאם, ולא להפך.

4. הטענה איננה נכונה, הדפדפנים אמנם רצים ב- user mode אך הם עדיין יכולים לבצע system calls.
5. הטענה אינה נכונה, תהליך אשר התחיל מתוך מערכת ההפעלה פועל בדר"כ ב- user mode. גם אלה שהותקנו עם מערכת ההפעלה.
6. הטענה אינה נכונה, ה- interrupts מגיעים ל- CPU ללא תלות במערכת ההפעלה. מערכת ההפעלה תוכל לבחור איך להגיב אליהם.
7. הטענה אינה נכונה. virtual machine משתמש במשאבים הקיימים כדי להריץ את מערכת ההפעלה שלו, ולא יכול לייעל את מהירות התפקוד של הרכיבים הקיימים.
8. הטענה אינה נכונה. ניתן לבנות כזאת תוכנה ב- user mode אשר מבוססת על קריאה מתוך ה- CD rom באמצעות system calls מתאימים.
9. הטענה אינה נכונה, כפי שראינו בתרגיל התכנות, system calls רבים יותר רק מאריכים את זמן הריצה.
10. הטענה לא נכונה, רכיבים חיצוניים יכולים ליצור interrupts בלבד. system calls נוצרים על ידי processes